THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Présentée par

Claire JONCHERY

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Spécialité : **Mathématiques**

Estimation d'un mouvement de caméra et problèmes connexes.

Thèse présentée et soutenue à Cachan le 6 novembre 2006 devant le jury composé de :

Xavier DESCOMBES	INRIA Sophia Antipolis	Rapporteur
Françoise DIBOS	Université Paris 13	Directrice de Thèse
Renaud KERIVEN	École Nationale des Ponts et Chaussées	Rapporteur
Georges KOEPFLER	Université Paris 5	Invité
Yves MEYER	École Normale Supérieure de Cachan	Président du jury
Lionel MOISAN	Université Paris 5	Examinateur
Jean-Michel MOREL	École Normale Supérieure de Cachan	Examinateur

Centre de Mathématiques et de leurs Applications
ENS CACHAN / CNRS / UMR 8536
61, avenue du Président Wilson, 94235 CACHAN CEDEX (France)

Remerciements

Je tiens à remercier en premier lieu Françoise Dibos, qui a dirigé mes recherches, pour sa patience, sa disponibilité et son enthousiasme pendant ces quatre années.

Xavier Descombes et Renaud Keriven ont eu la gentillesse d'accepter d'être les rapporteurs de cette thèse. Je les remercie pour leurs remarques et leurs suggestions qui m'orientent vers de nouvelles pistes de recherches.

Je suis très reconnaissante à Jean-Michel Morel et Lionel Moisan d'avoir accepté de faire partie de mon jury. Je remercie Yves Meyer de me faire l'honneur de le présider.

Je souhaite remercier chaleureusement Georges Koepfler pour l'attention constante qu'il a portée à mon travail depuis le début de mes recherches, toujours avec bonne humeur. Je remercie aussi pour leurs précieuses relectures et leurs remarques, Mathieu Lewin, Lionel Moisan, Julie Delon, Sophie Rainero, Sylvain Pelletier, Claire Lacour et Nabile Boussaïd.

Je n'aurais garde d'oublier les thésards du CEREMADE, Olivia, Mathieu, Nabile, Denis, Adrien, Sylvain, Florent, Fethallah et Pablo et la fameuse ambiance du C618. Je n'oublie pas non plus les doctorants du MAP5 qui m'ont accueillie pour la dernière année dans leur beau bureau, Amandine, Claire, Gwendoline, Cécile et Arno. Je remercie aussi Valérie et Sophie dont les encouragements, depuis le Texas et le Japon, me furent très précieux.

Merci enfin à toute ma famille, en particulier à mes parents pour leur indéfectible soutien pendant ces années de thèse et pendant toutes les autres, et bien sûr à Anne (jumelle jusque dans la thèse !).

Et un grand merci à Jacques-Olivier.

Table des matières

Notations

Image

- On considére seulement les images en niveaux de gris. Une image est représentée par une fonction continue ou discrète $I : D \to \mathbb{R}$, où D est un domaine borné de $\mathbb{R}^2$ ou de $\mathbb{Z}^2$. La valeur $I(x, y)$ correspond au niveau de gris de l'image au point (x, y).

Géométrie projective et euclidienne

- Les points de l'espace $\mathbb{R}^3$ sont notés en lettres majuscules, par exemple M.
- Les points du plan image sont notés en lettres minuscules, par exemple m.
- Le vecteur colonne de coordonnées **euclidiennes** associé à un point de $\mathbb{R}^3$ ou à un point du plan image, est noté par la même lettre (par exemple, M ou m). Il peut aussi être écrit sous sa forme développée (x, y) ou $\begin{pmatrix} x \\ y \end{pmatrix}$.
- Un vecteur colonne de coordonnées **projectives** associé à un point de l'espace tridimensionnel M ou de l'espace image m est noté par la même lettre mais en caractères gras, $\mathbf{M}$ ou $\mathbf{m}$.

Calcul matriciel

- On note $\langle u, v \rangle$ le produit scalaire euclidien de deux vecteurs de $\mathbb{R}^n$

$$\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i.$$

- On note $\|v\|$ la norme d'un vecteur de $\mathbb{R}^n$ définie à partir du produit scalaire euclidien

$$\|v\| = \sqrt{\langle v, v \rangle} = \sqrt{\sum_{i=1}^{n} v_i^2}.$$

- On note $\|A\|$ la norme de Frobenius d'une matrice de $\mathcal{M}_n(\mathbb{R})$

$$\|A\| = \sqrt{\sum_{i,j=1}^{n} A_{ij}^2}.$$

- On note A^T la transposée de la matrice A.
- Pour tout vecteur $v = (v_1, v_2, v_3)$ de $\mathbb{R}^3$, on note $[v]_\times$ la matrice antisymétrique

$$[v]_\times = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}.$$

  Ceci permet de remplacer le produit vectoriel par un produit matriciel ; pour tous vecteurs v et w de $\mathbb{R}^3$, $v \wedge w = [v]_\times w$.

Groupes

- On note $GL(n, \mathbb{R})$ l'ensemble des matrices carrées réelles d'ordre n inversibles, $SL(n, \mathbb{R})$ l'ensemble des matrices de $GL(n, \mathbb{R})$ de déterminant égal à 1 et $SO(n)$ l'ensemble des matrices de $GL(n, \mathbb{R})$ orthogonales de déterminant 1.
- Les groupes sont notés en lettres majuscules et les algèbres en lettres gothiques minuscules, par exemple : le groupe $SO(3)$ et son algèbre $\mathfrak{so}(3)$.

Notations de Landau

- On note $g = o(f)$ quand $x \to a$ si et seulement si

$$\forall \varepsilon > 0, \quad \exists\, \eta > 0 \quad \text{tel que} \quad |x - a| < \eta \;\Rightarrow\; |g(x)| < \varepsilon\, |f(x)|.$$

Fonctions

- On note sgn la fonction signe, définie sur $\mathbb{R}$ par

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0. \end{cases}$$

Introduction

Acquisition d'images par une caméra

Lorsqu'un appareil photographique numérique prend une photo, il convertit une vue du monde tridimensionnel en une image numérique bidimensionnelle. Le système d'acquisition de l'appareil contient un ou plusieurs capteurs qui transforment les photons en un signal électrique ; celui-ci est ensuite numérisé par un convertisseur analogique/digital puis traité pour obtenir une image numérique. C'est l'étape de numérisation qui définit la résolution de l'image, c'est-à-dire le nombre de pixels par unité de longueur. Ainsi, une image numérique I est une fonction discrète où $I(x, y)$ représente le niveau de gris sur l'image au pixel (x, y). Le procédé classique de formation des images est le modèle sténopé appelé aussi modèle pinhole dans la littérature anglo-saxonne ; l'appareil réalise une projection centrale de l'espace $3D$ sur une surface, comme illustré sur la figure (1).

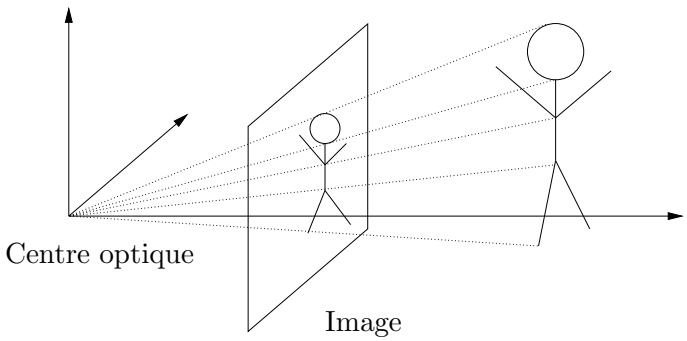

FIGURE 1: *Formation des images par le modèle pinhole.*

Lorsqu'une caméra se déplace dans l'espace, elle acquiert 24 images par seconde, par le procédé décrit ci-avant. Si la scène filmée est statique, il existe alors des relations, d'une part entre la scène 3D filmée et les différentes vues de la scène, d'autre part entre les images de la séquence. Ces relations dépendent du mouvement de la caméra et de la structure de la scène. Elles sont décrites dans le chapitre 1.

Nous étudions dans cette thèse trois problèmes liés au mouvement d'une caméra dans un environnement statique. Le premier est l'estimation du mouvement de la caméra filmant une scène fixe à partir de la séquence d'images obtenues, le deuxième est l'estimation des profondeurs

de la scène à partir du mouvement estimé. Le troisième concerne l'unicité du mouvement et de la structure de la scène à estimer : on s'interroge sur l'injectivité de la fonction associant un flot optique à un mouvement de caméra et à la structure de la scène.

Motivations à l'estimation d'un mouvement de caméra

La connaissance du mouvement d'une caméra a de nombreuses applications. Elle est à la base des méthodes appelées "structure from motion", méthodes estimant un plan des profondeurs de la scène filmée (ou structure) à partir des images du film et du mouvement de la caméra.

Elle est aussi utilisée pour la compensation de mouvement entre deux images, opération très efficace en compression pour diminuer le coût de codage d'une vidéo [79], ou pour stabiliser une séquence d'images.

Un autre champ d'applications important est la réalité augmentée. L'idée est d'ajouter à des images d'un monde réel des objets virtuels [40]. La connaissance du mouvement de la caméra est alors nécessaire pour insérer les objets virtuels dans la séquence avec le même point de vue que celui adopté par la caméra lors du tournage. En médecine, cette technique permet d'assister le chirurgien pendant une opération en superposant un modèle de l'organe opéré à la vision réelle donnée par un endoscope par exemple. En urbanisme, la réalité augmentée permet par exemple d'examiner des projets de construction en insérant un futur bâtiment dans une vidéo tournée sur le site d'implantation. Les domaines d'application sont nombreux : cinéma, architecture d'intérieur... La figure (2) présente un exemple de réalité augmentée simplifiée car l'objet inséré (une affiche de cinéma) est plan. L'affiche originale est insérée dans une première image puis, à l'aide du mouvement de caméra, elle est déformée pour être ajoutée dans les images suivantes de la séquence.

Résumé par chapitres

Le chapitre 1 de la thèse est consacré à la description de modèles, d'outils et de méthodes existants. On y décrit la projection réalisée par une caméra lors de l'acquisition d'images, différentes modélisations d'une rotation dans l'espace et un mouvement quelconque de caméra. On expose ensuite les relations entre deux vues d'une scène fixe ; s'il n'y a pas d'effet de parallaxe, on peut apparier les points des images deux à deux, sinon, les images sont liées par la contrainte épipolaire. Enfin, on présente une revue succinte des méthodes d'estimation du mouvement d'une caméra dans un environnement statique.

Dans le chapitre 2, on considère deux images consécutives dans une séquence. Les images des couples étudiés diffèrent donc très peu : les bords des objets de la scène apparaissent essentiellement sur l'image des différences (figure (3)). Deux points de ces images appariés, c'est-à-dire projections d'un même point de l'espace, sont liés par des applications dépendant du mouvement de la caméra et de la profondeur du point de l'espace projeté. Pour s'affranchir de cette profondeur variable suivant les appariements, on définit un contexte permettant d'approximer la profondeur de la scène par une constante, dans les applications liant les deux images. On montre

FIGURE 2: *Exemple de réalité augmentée. En haut, le tableau d'affichage du bureau est masqué par une affiche de cinéma. Au-dessous, les images* 10, 20 *et* 30 *de la séquence obtenue en déformant l'affiche avec le mouvement de la caméra.*

que si le produit de l'amplitude des variations de l'inverse de la profondeur par la norme de la translation est suffisamment faible, et si la caméra est suffisamment éloignée de la scène, on peut approximer les applications liant les deux vues par des applications projectives ne dépendant que des paramètres du mouvement de la caméra. On propose de modéliser ces déformations, non pas dans le groupe projectif, mais dans le groupe des recalages [11], isomorphe au groupe des déplacements dans l'espace. Ainsi, la composition et l'inversion des déformations seront toujours associées à des mouvements de caméra. On présente ensuite une décomposition originale du mouvement de la caméra permettant de séparer la déformation projective entre deux images consécutives en deux composantes : une similitude et une déformation "purement" projective. Cette nouvelle écriture conduit à une approximation quadratique du flot optique entre deux images consécutives, qui met en évidence les régions de l'image déformées par l'une ou l'autre des composantes du mouvement.

Dans le chapitre 3, on propose une méthode d'estimation du mouvement entre deux images consécutives. Cette méthode est basée sur l'approximation quadratique obtenue dans le chapitre 2. Elle utilise l'approche d'Odobez et Bouthémy d'estimation de mouvements bidimensionnels entre deux images [51], implémentée dans le logiciel Motion2D. L'utilisation du logiciel a nécessité l'ajout d'un modèle de mouvement à six paramètres, correspondant à l'approximation

quadratique. La méthode proposée est rapide, robuste, et présente les avantages inhérents aux méthodes directes : elle ne nécessite ni calcul de flot optique ni appariement de points préalablement à son application. Ses performances sont illustrées à travers les estimations de mouvement obtenues sur des films synthétiques et réels, et quelques utilisations de ces estimations, comme le mosaïquage. On montre de plus que la méthode peut s'appliquer hors des limites du cadre fixé, par exemple quand la caméra s'approche de la scène ou quand un objet a un mouvement propre.

FIGURE 3: *Deux images consécutives issues d'une séquence réelle et l'image des différences entre les deux (plus le niveau de gris est foncé, plus la différence est importante). Deux problèmes sont examinés : l'estimation du mouvement de la caméra à partir d'un couple d'images consécutives, puis l'estimation de la structure de la scène.*

L'objet du chapitre 4 est l'utilisation de l'estimation du mouvement de la caméra entre deux images consécutives pour déterminer la structure de la scène filmée. Nous appliquons pour cela une méthode de Belief Propagation, déjà utilisée à cette fin en stéréovision par Sun, Shum et Zheng dans [67]. À la différence des approches existantes, nous mettons en oeuvre cette méthode probabiliste directement sur un couple d'images consécutives, sans rectification, en utilisant le mouvement estimé. Ce mouvement permet l'initialisation d'une carte des profondeurs de la scène en fournissant en chaque pixel de la première image une distribution de probabilité sur les profondeurs du point de l'espace projeté en ce pixel. Pour améliorer les résultats d'estimation à la fois des profondeurs et du mouvement, nous proposons d'évaluer un nouveau mouvement entre les images en tenant compte du plan de profondeurs estimé, puis d'itérer le procédé.

Le chapitre 5 est le fruit d'une collaboration avec J.-O. Moussafir de Saint-Gobain Recherche. Nous examinons l'injectivité de l'application qui associe au mouvement d'une caméra et à la structure de la scène filmée, le flot optique correspondant. Ce problème est traité sous l'angle du domaine d'observation du flot optique. Nous prouvons que l'application étudiée est injective si le flot optique est observé, dans le cas d'une projection sténopé, sur le plan rétinien tout entier. Nous nous intéressons ensuite aux contre-exemples ; deux mouvements de caméra étant donnés, nous décrivons le domaine d'observation où les flots générés sont susceptibles d'être identiques et déduisons les équations des surfaces filmées qui, associées à ces deux mouvements de caméra,

produiront le même flot optique sur ce domaine.

Chapitre 1

Estimation d'un mouvement de caméra : outils et état de l'art

Ce chapitre introductif est consacré à la description de notre cadre de travail ; nous présentons d'abord le modèle mathématique de la projection de l'espace sur un plan réalisée par la caméra puis le modèle de mouvement d'une caméra dans l'espace tridimensionnel. Nous évoquons ensuite les liens entre le flot optique et la vitesse de la caméra, puis les relations entre deux vues d'une scène fixe. Enfin, nous terminons par un état de l'art succinct des méthodes d'estimation du mouvement d'une caméra filmant une scène fixe.

1.1 Présentation du modèle projectif

Une caméra fournit des images planes d'un monde que nous percevons en trois dimensions. Elle utilise donc un procédé de projection, appelé fonction de projection, qui associe à un point de l'espace tridimensionnel sa projection dans l'image donnée par la caméra. Cette fonction de projection n'est en général pas quelconque et appartient à une famille de fonctions dépendant du modèle de caméra choisi. Le modèle le plus courant et que nous allons utiliser est le modèle projectif linéaire ou sténopé.

Dans ce modèle, la caméra réalise une projection centrale de l'espace euclidien sur une surface : c'est le principe de l'appareil photographique à trou d'épingle.

Comment décrire mathématiquement la projection réalisée par la caméra ? Il est tout d'abord nécessaire de décrire l'espace à trois dimensions et pour cela de choisir une géométrie adaptée à la projection réalisée par la caméra. Félix Klein, au début du vingtième siècle, caractérise les géométries par des groupes de transformations, qui laissent invariantes certaines propriétés des objets de l'espace. La géométrie euclidienne par exemple est caractérisée par le groupe des transformations euclidiennes, les rotations et les translations, qui laissent invariants les rapports des distances et les angles. Elle est bien adaptée à notre perception du monde 3D car elle rend parfaitement compte de l'expérience que nous en avons. Cependant, la projection réalisée par

une caméra n'appartient pas au groupe des transformations euclidiennes ; le parallélisme de deux droites, par exemple, n'est pas préservé par la projection car les droites projetées sur une image s'intersectent en un point de la ligne d'horizon. Elle est décrite par la géométrie projective, extension de la géométrie euclidienne. Le groupe de transformations associé à cette géométrie ne conserve que le birapport de points de l'espace mais contient davantage de transformations, dont la projection réalisée par une caméra.

L'élaboration de la géométrie projective s'inscrit dans une perspective historique ; la majorité des résultats datent de la Grèce Antique et de la Renaissance. Les Grecs Anciens découvrirent de nombreuses propriétés géométriques de la projection, comme la conservation du birapport et la notion de mouvement de parallaxe. Au quinzième siècle, en Europe, la volonté de réalisme pictural rendit nécessaire l'introduction d'une nouvelle méthode de représentation : la perspective (ou projection centrale). L'architecte florentin Brunelleschi décrivit le premier les règles de la perspective, règles reprises et développées notamment par Alberti, Della Francesca, Dürer et Vinci. Les siècles suivants, de nombreux mathématiciens, Desargues, Pascal, Monge, Poncelet... entre autres, contribuèrent au développement de ces notions. Dans les années 80, avec le développement des ressources informatiques, la géométrie projective est naturellement devenue un outil de premier plan pour la résolution de problèmes de vision par ordinateur. Faugeras [13, 15], Kanatani [38], Hartley et Zisserman [26] ont en particulier écrit des ouvrages de référence sur le sujet.

1.1.1 Modèle sténopé

Décrivons maintenant le modèle projectif, dit aussi sténopé, de caméra. Le modèle projectif est défini par deux éléments : un point, appelé centre optique et un plan ne contenant pas le point, appelé plan rétinien. La figure (1.1) présente ces deux éléments. Le centre optique C ou centre de projection correspond à la position de la caméra et le plan rétinien $\mathcal{R}$ est le plan de formation des images. On appelle longueur focale f_c la distance strictement positive du point C au plan $\mathcal{R}$, rayon optique toute demi-droite d'extrémité C intersectant le plan $\mathcal{R}$, et axe optique l'axe orthogonal à $\mathcal{R}$ passant par C. Il intersecte $\mathcal{R}$ en un point c appelé point principal.

Soit π la fonction de projection de l'espace dans le plan rétinien

$$\begin{array}{rccc} \pi : & \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ & M = (X, Y, Z) & \longmapsto & m = (x, y). \end{array}$$

L'image m par π d'un point M de l'espace est l'intersection du rayon optique CM avec le plan $\mathcal{R}$. L'expression de la fonction de projection dépend des systèmes de coordonnées utilisés dans l'espace et dans le plan image.

Considérons le repère orthonormal de $\mathbb{R}^3$ (C, i, j, k), d'origine le centre optique C de la caméra et tel que k soit un vecteur directeur de l'axe optique. Ce repère est appelé système de coordonnées standard de la caméra. Munissons maintenant le plan rétinien $\mathcal{R}$ du repère orthonormal bidimensionnel (c, i, j), d'origine le point principal du plan rétinien et dont les axes

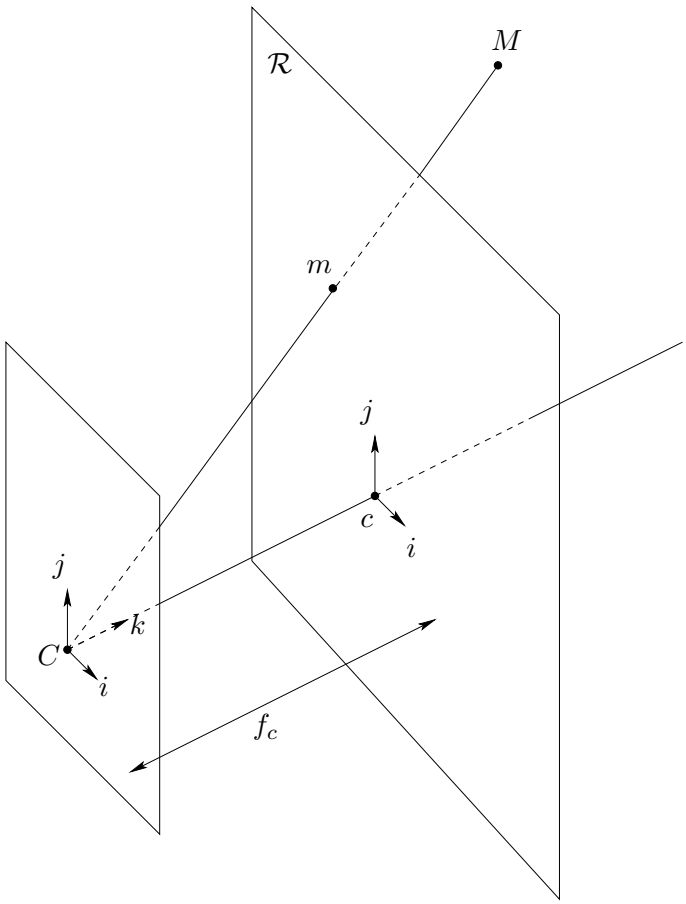

FIGURE 1.1: *Modèle de caméra sténopé.*

de coordonnées sont parallèles à ceux du repère associé à la caméra. Dans ces deux systèmes de coordonnées, l'expression de la fonction π résulte de l'application du théorème de Thalès

$$\begin{cases} x = f_c \, \dfrac{X}{Z} \\ \\ y = f_c \, \dfrac{Y}{Z}. \end{cases} \tag{1.1}$$

Remarque – Deux triplets de coordonnées proportionnels de l'espace sont projetés en un même point du plan $\mathcal{R}$. Tous les points M d'un rayon optique ont ainsi une projection unique m sur le plan rétinien : il est impossible de déterminer les profondeurs Z des points projetés sur une image à partir de cette seule image.

La remarque précédente permet d'introduire les coordonnées projectives. Si (x, y) sont les coordonnées euclidiennes d'un point sur un plan, ses coordonnées projectives (ou homogènes)

sont définies par $(\lambda x : \lambda y : \lambda)$, pour tout réel λ non nul. L'espace des triplets de coordonnées, avec la règle que les triplets proportionnels représentent le même point, est appelé plan projectif.

Définition 1.1 – *On appelle plan projectif le quotient de* $\mathbb{R}^3 \setminus \{0\}$ *par la relation d'équivalence*

$$(X, Y, Z) \cong (X', Y', Z') \Leftrightarrow \exists \lambda \neq 0 \text{ tq } (X, Y, Z) = \lambda(X', Y', Z').$$

On note $\mathbf{M} = (X : Y : Z : 1) = (\lambda X : \lambda Y : \lambda Z : \lambda)$ les coordonnées projectives d'un point M de coordonnées euclidiennes (X, Y, Z) de $\mathbb{R}^3$ et $\mathbf{m} = (x : y : 1) = (\lambda x : \lambda y : \lambda)$ celles d'un point m de coordonnées euclidiennes (x, y) dans le plan rétinien.

1.1.2 Matrice de projection

La fonction de projection π définie par l'expression (1.1) de $\mathbb{R}^3$ dans $\mathbb{R}^2$ n'est pas linéaire par rapport aux coordonnées euclidiennes X, Y et Z. Mais elle le devient si on utilise les coordonnées projectives dans l'espace et dans le plan rétinien. Considérons les systèmes de coordonnées homogènes associés au repère (C, i, j, k) et au repère (c, i, j) respectivement pour l'espace des objets et pour l'image. Soit M un point de l'espace, de coordonnées euclidiennes (X, Y, Z) dans le repère (C, i, j, k). Il a pour coordonnées homogènes, non uniques, $(X' : Y' : Z' : T')$ avec $X = X'/T', Y = Y'/T', Z = Z'/T'$. De même, le point m de l'image, de coordonnées euclidiennes (x, y) dans (c, i, j) a pour coordonnées homogènes non uniques $(x' : y' : z')$ avec $x = x'/z', y = y'/z'$. La relation (1.1) peut alors s'écrire

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X/Z \\ Y/Z \\ 1 \\ 1/Z \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} Zx \\ Zy \\ Z \end{pmatrix} = \begin{pmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} Zx \\ Zy \\ Z \end{pmatrix} = \mathcal{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1.2}$$

Comme $\mathbf{m} = (Zx : Zy : Z)$ et $\mathbf{M} = (X : Y : Z : 1)$, on obtient

$$\mathbf{m} = \mathcal{P}\,\mathbf{M}. \tag{1.3}$$

L'écriture de la fonction de projection est ainsi facilitée par l'utilisation de la géométrie projective.

Définition 1.2 – *On appelle matrice de projection $\mathcal{P}$ associée à une caméra la matrice de dimensions 3×4 représentant la projection des points de $\mathbb{R}^3$ sur le plan rétinien en coordonnées projectives*

$$\mathbf{m} = \mathcal{P}\,\mathbf{M}.$$

La matrice $\mathcal{P}$ est de rang 3 et est définie à un scalaire près. Le centre optique C est l'unique point vérifiant $\mathcal{P}\mathbf{C} = 0$. Suivant les systèmes de coordonnées choisis dans l'espace et dans le plan rétinien, la matrice de projection s'écrit différemment.

1.1.3 Paramètres intrinsèques et extrinsèques de la caméra

En choisissant le système de coordonnées standard de la caméra (C, i, j, k) comme repère de l'espace et le système (c, i, j) comme repère de l'image, l'écriture de la matrice de projection $\mathcal{P}$ était particulièrement simple. Mais il est possible que le système de coordonnées choisi dans $\mathbb{R}^3$ ne soit pas celui de la caméra et que le repère choisi dans l'image ne corresponde pas à (c, i, j), comme illustré sur la figure (1.2). La projection d'un point M sur le plan rétinien $\mathcal{R}$ revient alors à faire un changement de repère dans l'espace, du repère donné à (C, i, j, k), la projection de (C, i, j, k) dans (c, i, j) et un changement de repère dans l'image, de (c, i, j) au repère choisi.

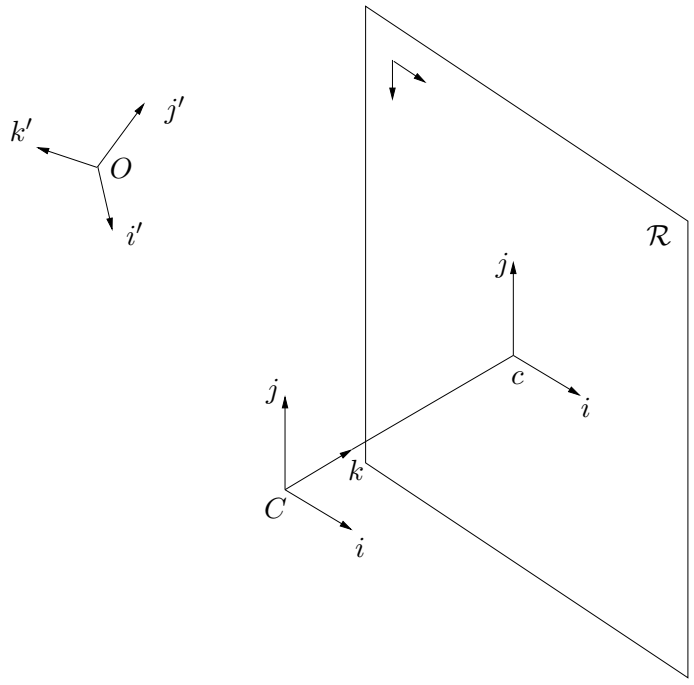

FIGURE 1.2: *Différents systèmes de coordonnées dans l'espace et dans l'image.*

Soit (O, i', j', k') le repère orthonormal de l'espace dans lequel sont repérées les positions des objets dans $\mathbb{R}^3$. Le changement de repère de (O, i', j', k') à (C, i, j, k) est un déplacement dans $\mathbb{R}^3$, que l'on peut décomposer en une rotation R suivie d'une translation de vecteur t. Si un point M de l'espace a pour coordonnées M_O et M_C dans les deux systèmes de coordonnées, alors

$$M_O = RM_C + t,$$

où le vecteur de translation t est égal au vecteur OC et la rotation R est définie par $R(i) = i'$, $R(j) = j'$ et $R(k) = k'$. On appelle paramètres extrinsèques de la caméra, l'expression de la position et de l'orientation de la caméra dans (O, i', j', k'), c'est-à-dire la matrice R et le vecteur t.

Décrivons maintenant le changement de repère sur l'image. Le système d'acquisition de la caméra contient un capteur formé de cellules photosensibles qui génèrent chacune un point lumineux, appelé pixel sur l'image. La caméra fournit l'image sous forme d'un tableau à deux dimensions, dont les valeurs sont les niveaux de gris des pixels. Le système de coordonnées associé à l'image, appelé système de coordonnées pixels, est propre à la caméra, et souvent différent du repère (c, i, j). On appelle paramètres intrinsèques de la caméra les paramètres reliant le système de coordonnées pixels au repère (c, i, j) sur le plan rétinien. Le passage des coordonnées aux pixels est réalisé par un changement de repère affine

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha_u x + \gamma y + u_0 \\ \alpha_v y + v_0 \end{pmatrix} = \begin{pmatrix} \alpha_u & \gamma \\ 0 & \alpha_v \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}.$$

Les facteurs multiplicatifs α_u et α_v expriment la longueur focale en pixels sur chacun des axes. Ils sont souvent donnés par les dimensions des pixels sur le capteur de la caméra par le constructeur. Le couple (u_0, v_0) exprime les coordonnées du point principal c, qui sont rarement $(0, 0)$ car l'origine du repère des pixels est souvent localisée en un coin de l'image. Enfin, le paramètre γ est en général égal à zéro, ou très proche de zéro car pour la plupart des caméras, les cellules photosensibles du capteur sont rectangulaires.

En tenant compte des paramètres intrinsèques et extrinsèques de la caméra, la matrice de projection d'un point M dans le repère (O, i', j', k') sur le point m dans le repère des pixels s'écrit

$$\mathcal{P} = A\,\mathcal{P}_0\,K \tag{1.4}$$

où

$$A = \begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathcal{P}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{et} \qquad K = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}.$$

Les matrices A et K sont respectivement celles des paramètres intrinsèques et extrinsèques. La longueur focale f_c n'apparaît pas dans la matrice de projection $\mathcal{P}_0$ car elle est exprimée dans les facteurs α_u et α_v.

Connaissant la matrice de projection $A\mathcal{P}_0K$, on peut, à partir des coordonnées d'un pixel, localiser le rayon optique, c'est-à-dire la droite de l'espace passant par C, à laquelle appartient le point projeté en ce pixel. Si seule la matrice A est connue, la position $3D$ du rayon optique correspondant à un pixel donné, sera localisée dans le repère de la caméra. Dans ce cas, la caméra est dite calibrée. De nombreuses méthodes de calibrage de caméra existent [14] ; la méthode classique consiste à utiliser des appariements de points de l'espace 3D de coordonnées connues, avec des points de l'image pour déterminer les 5 ou 11 paramètres de la matrice (5 pour la matrice A seule et 11 pour les matrices A et K).

Cependant, il n'est pas toujours nécessaire de calibrer la caméra ; tant qu'aucune interprétation 3D n'est faite, le modèle non calibré suffit.

1.1.4 Rôle de la longueur focale

Revenons sur les formules (1.1). La longueur focale f_c agit comme un facteur d'échelle sur l'image. Ainsi, on peut poser en toute généralité $f_c = 1$. Cela revient à choisir f_c comme unité du système de coordonnées de la caméra (C, i, j, k) ; le système standard de coordonnées de la caméra est alors dit normalisé.

Dans la suite du document, les paramètres intrinsèques de la caméra sont supposés connus ; nous prendrons la longueur focale égale à 1 et la matrice A égale à I_3. Dans les expériences, l'origine sur les images étant localisée au coin en haut à gauche, la connaissance des dimensions des images et donc de (u_0, v_0) modifiera les formules.

1.2 Modélisation d'un mouvement de caméra

1.2.1 Modélisation d'une rotation

Une rotation de $\mathbb{R}^3$ est une matrice 3×3 orthogonale de déterminant égal à 1. L'ensemble de ces matrices forme le groupe Spécial Orthogonal, noté $SO(3)$. Plusieurs paramétrages de ces rotations existent.

1.2.1.1 Représentation canonique

La représentation habituelle consiste à définir une rotation à l'aide de trois paramètres : l'axe de rotation représenté par un vecteur unitaire u (soit un point sur S^2, la sphère unité de $\mathbb{R}^3$, déterminé par deux paramètres) et l'angle de rotation a autour de l'axe, comme illustré sur la figure (1.3).

Définition 1.3 – *On note $\mathcal{A}$ l'ensemble des matrices antisymétriques*

$$[u]_\times = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}$$

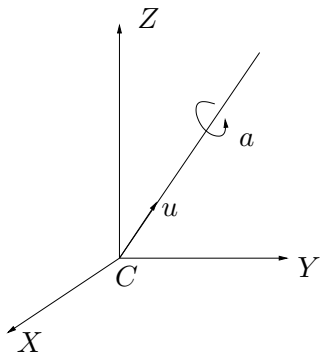

FIGURE 1.3: *Représentation canonique d'une rotation.*

vérifiant $\|u\| = \sqrt{u_1^2 + u_2^2 + u_3^2} = 1$.

Comme $\mathcal{A}$ et S^2 sont en bijection, on peut indifféremment définir l'axe de la rotation par un point de S^2 ou une matrice de $\mathcal{A}$.

Proposition 1.1 – *Soit une rotation définie par le vecteur directeur unitaire u de son axe et son angle a. L'application associant à u et a la matrice de rotation est donnée par la formule de Rodrigues (1817)*

$$\begin{array}{rcl} [0, 2\pi[\times\mathcal{A} & \rightarrow & SO(3) \\ (a, [u]_\times) & \mapsto & R = I_3 + \sin a\ [u]_\times + (1 - \cos a)\ ([u]_\times)^2 \end{array} \tag{1.5}$$

Cette application est surjective mais non injective. En restreignant l'ensemble de départ à $]0, \pi[\times S^2 \cup \{(0, (0, 0, 1))\}$*, on a l'application inverse*

$$SO(3) \rightarrow\ ]0, \pi[\times\mathcal{A} \cup \{(0, [(0, 0, 1)]_\times)\}$$

$$R \mapsto \begin{cases} \left(\arccos\left(\frac{tr(R)-1}{2}\right), \frac{1}{2\sin\left(\arccos\left(\frac{tr(R)-1}{2}\right)\right)}(R - R^T)\right) & si\ tr(R) \neq 3 \\ \left(0, \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right) & si\ tr(R) = 3. \end{cases}$$

Démonstration. L'application donnée par la formule de Rodrigues n'est pas injective car les rotations définies par $(a, [u]_\times)$ et $(-a[2\pi], [-u]_\times)$ donnent la même matrice de rotation. Il faut donc restreindre les valeurs des angles de rotation à $[0, \pi[$. De plus, $\forall u \in S^2$, la matrice de rotation définie par $(0, [u]_\times)$ sera toujours égale à I_3 ; en restreignant l'espace de départ à $]0, \pi[\times S^2 \cup \{(0, (0, 0, 1))\}$, la fonction devient bijective et la solution au problème inverse est obtenue en remarquant que $tr(R) = 2\cos a + 1$ et $R - R^T = 2\sin a\ [u]_\times$. □

1.2.1.2 Quaternions

Définition 1.4 – *On appelle quaternion tout quadruplet de réels $Q = (q_0, q_1, q_2, q_3)$, que l'on note aussi*

$$Q = (q_0, q) \text{ avec } q = (q_1, q_2, q_3).$$

L'ensemble des quaternions, muni du produit

$$PQ = R \text{ avec } \begin{cases} r_0 = p_0\, q_0 - \langle p, q \rangle \\ r = p_0\, q + p\, q_0 + p \wedge q \end{cases}$$

forme une algèbre à 4 dimensions. Le produit scalaire sur l'algèbre des quaternions est donné par

$$P.Q = p_0\, q_0 + \langle p, q \rangle.$$

- *Si $Q.Q = 1$, on dit que Q est unitaire.*
- *On note le conjugué de Q, $Q^* = (q_0, -q)$.*

Montrons qu'il existe une application de l'ensemble des quaternions unitaires dans le groupe des rotations [58].
Soit $X = (x_0, x)$ un quaternion et $Q = (q_0, q)$ un quaternion unitaire. On considère la transformation $X' = QXQ^*$, alors

$$\begin{cases} x'_0 = x_0 \\ \\ x' = (q_0^2 - q.q)\, x + 2q_0\, q \wedge x + 2q\, \langle q, x \rangle. \end{cases}$$

Comme Q est unitaire, il existe un réel a appartenant à $[-\pi, \pi[$ et un vecteur u de $\mathbb{R}^3$ de norme 1 tel que

$$Q = \left(\cos \frac{a}{2}, \sin \frac{a}{2}\, u \right).$$

Alors, comme $q\,(q.x) = (\sin \frac{a}{2})^2\, u\, u^T\, x = (\sin \frac{a}{2})^2\, ([u]_\times^2 + I_3)\, x$, on a

$$\begin{aligned} x' &= ((\cos \tfrac{a}{2})^2 - (\sin \tfrac{a}{2})^2)\, x + 2 \cos \tfrac{a}{2}\, \sin \tfrac{a}{2}\, [u]_\times\, x + 2\, (\sin \tfrac{a}{2})^2\, ([u]_\times^2 + I_3)\, x \\ &= x + \sin a\, [u]_\times\, x + (1 - \cos a)\, [u]_\times^2\, x \\ &= \left(I_3 + \sin a\, [u]_\times + (1 - \cos a)\, [u]_\times^2 \right)\, x. \end{aligned}$$

On retrouve la formule de Rodrigues : à tout quaternion unitaire, on peut associer une rotation.

Proposition 1.2 – *À tout quaternion unitaire $Q = (q_0, q_1, q_2, q_3)$, on peut associer une rotation d'angle a et d'axe dirigé par le vecteur unitaire u*

$$Q \longmapsto \begin{cases} a = 2\arccos(q_0) \\ u = \begin{cases} \dfrac{1}{\sin(\arccos q_0)} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} & si\ q_0 \neq \pm 1 \\ 0 & sinon. \end{cases} \end{cases}$$

La matrice de rotation R correspondante s'écrit

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(-q_0q_3 + q_1q_2) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(-q_0q_1 + q_2q_3) \\ 2(-q_0q_2 + q_1q_3) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}.$$

L'application qui à un quaternion unitaire associe une rotation n'est pas bijective car les quaternions Q et $-Q$ fournissent la même rotation.

La représentation des rotations par des quaternions est très utilisée en infographie. Son principal intérêt réside dans la composition des rotations ; en effet, la composition de rotations correspond à la multiplication des quaternions associés. Il peut être avantageux de préférer les quaternions aux matrices de rotations car la complexité de la multiplication des quaternions est plus faible que celle des rotations : alors que le produit de 2 matrices de rotations nécessite 27 multiplications et 18 additions, le produit des quaternions est calculé en seulement 16 multiplications et 12 additions. De plus, lors de la composition de rotations, des erreurs d'arrondis peuvent conduire à l'obtention d'un quaternion non unitaire. Il suffit alors de le normaliser. Le cas est plus compliqué pour la multiplication matricielle. Si une matrice A quasi orthogonale est obtenue, la réorthogonalisation nécessite le calcul de $A(A^TA)^{-1/2}$ [58].

1.2.1.3 Angles d'Euler

Une autre écriture consiste à décomposer une rotation en trois rotations autour de chacun des axes du repère. Soit α l'angle de rotation autour de l'axe (CX), β autour de (CY) et γ autour de (CZ). Ces trois angles sont appelés les angles d'Euler de la rotation.

Si on note $R(\alpha, \beta, \gamma)$ une telle rotation, elle s'écrit

$$R(\alpha, \beta, \gamma) = R(0, 0, \gamma) \circ R(0, \beta, 0) \circ R(\alpha, 0, 0)$$

soit matriciellement

$$\begin{pmatrix} \cos\gamma\cos\beta & -\cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & -\cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & -\sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & -\sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha \\ \sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{pmatrix}.$$

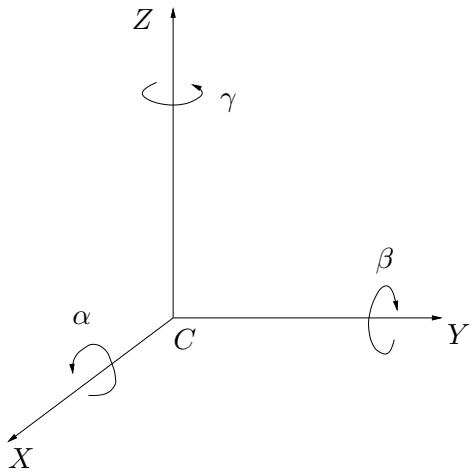

FIGURE 1.4: *Représentation d'une rotation par les angles d'Euler.*

Toutefois, l'application qui associe aux angles d'Euler (α, β, γ) la rotation de matrice $R(\alpha, \beta, \gamma)$ n'est pas injective ; par exemple, la rotation $R(\pi, \pi, 0)$ est égale à la rotation $R(0, 0, \pi)$.

Proposition 1.3 – *Une rotation définie par le vecteur directeur unitaire u de son axe et d'angle a a pour angles d'Euler $(\alpha, \beta, \gamma) = au$.*

Dans le chapitre 2, nous proposerons un nouveau paramétrage des rotations, lié aux types de transformations observées sur une image lors de rotations de caméra.

1.2.1.4 Exponentielle d'une matrice antisymétrique

Le groupe $SO(3)$ a la propriété d'être un groupe de Lie. Cette propriété permet d'associer une rotation de caméra à une vitesse angulaire. Commençons par la définition d'un groupe de Lie de matrices, voir par exemple Hall [24].

Définition 1.5 – *Un groupe de Lie de matrices réelles est un sous-groupe G de $GL(n, \mathbb{R})$ ayant la propriété suivante : si $(A_m)_{m\in\mathbb{N}} \in G$ et $\lim_{m\to+\infty} A_m = A$, alors $A \in G$ ou A n'est pas inversible.*

En conséquence, tout sous-groupe fermé de $GL(n, \mathbb{R})$ (ensemble des matrices $n \times n$ inversibles), est un groupe de Lie. C'est le cas de $SO(3)$, sous-groupe fermé de $GL(3, \mathbb{R})$.

Il existe plusieurs façons équivalentes de définir l'algèbre de Lie d'un groupe de Lie. Historiquement, les éléments de l'algèbre de Lie étaient vus comme les éléments infinitésimaux du groupe. Pour les groupes de matrices, l'algèbre de Lie est définie par l'intermédiaire de la fonction exponentielle. Commençons par rappeler la définition de l'exponentielle de matrice.

Définition 1.6 – *La fonction exponentielle de matrice est définie de $\mathcal{M}_n(\mathbb{R})$ dans $\mathcal{M}_n(\mathbb{R})$*

par le développement en séries de Taylor de l'exponentielle

$$\exp(A) = e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

La somme converge quelle que soit la matrice A.

Remarque – Soient A et $B \in \mathcal{M}_n(\mathbb{R})$. Si A et B commutent, alors

$$e^A e^B = e^{A+B}.$$

Une conséquence de la remarque est que l'exponentielle d'une matrice antisymétrique est une matrice orthogonale de déterminant égal à 1. En effet, si A est antisymétrique, $A^T = -A$ donc $e^{A^T} = e^{-A} (= (e^A)^T)$. Or, $e^A e^{-A} = I_n$ car A et $-A$ commutent, d'où $e^{-A} = (e^A)^{-1}$ et $(e^A)^T = (e^A)^{-1}$. De plus, $\det(e^A) = e^{tr(A)} = e^0 = 1$.

Définissons maintenant l'algèbre de Lie d'un groupe de Lie de matrices.

Définition 1.7 – *Soit G un groupe de Lie de matrices. L'algèbre de Lie de G, notée $\mathfrak{g}$, est l'ensemble de toutes les matrices M telles que $\forall t \in \mathbb{R}$, $e^{tM} \in G$.*

Nous avons vu précédemment que l'exponentielle d'une matrice antisymétrique est une matrice orthogonale de déterminant égal à 1. La réciproque est aussi vraie [24]. Ainsi, l'algèbre de Lie $\mathfrak{so}(3)$ associée au groupe $SO(3)$ est l'ensemble des matrices antisymétriques de $\mathcal{M}_3(\mathbb{R})$.

$$\begin{aligned}
\mathfrak{so}(3) &= \left\{ [\omega]_\times = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix},\ \omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3 \right\} \\
&= \text{Vect}\left\{ L_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, L_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, L_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\}.
\end{aligned}$$

Remarque – Soit $M \in \mathfrak{so}(3)$. Comme e^{tM} est une rotation, $\frac{d}{dt} e^{tM}\Big|_{t=0}$ est une rotation infinitésimale. Or, $\frac{d}{dt} e^{tM}\Big|_{t=0} = M$, donc l'algèbre $\mathfrak{so}(3)$ est isomorphe à l'ensemble des rotations infinitésimales, c'est-à-dire à l'ensemble des vitesses angulaires $(\omega_1, \omega_2, \omega_3)$ autour des axes (CX), (CY) et (CZ) associées aux rotations.

Proposition 1.4 – *La fonction*

$$\begin{array}{rccc} \exp : & \mathfrak{so}(3) & \to & SO(3) \\ & A & \mapsto & e^A \end{array}$$

est surjective et non injective.
Précisément, si

$$A = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

alors e^A *est une matrice de rotation d'axe dirigé suivant le vecteur unitaire* $u = \dfrac{(\omega_1, \omega_2, \omega_3)}{\sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2}}$ *et d'angle* $a = \sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2}\ [2\pi]$.

Démonstration. 1) Soit

$$A = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}.$$

Montrons que e^A est une matrice de rotation d'axe dirigé suivant le vecteur $\omega = (\omega_1, \omega_2, \omega_3)$ et d'angle $\|\omega\|$. Comme $A^3 = -\|\omega\|^2 A$,

$$\begin{aligned} e^A &= \sum_{k=0}^{\infty} \frac{A^k}{k!} \\ &= I_3 + \sum_{k=0}^{\infty} \frac{(-1)^k \|\omega\|^{2k}}{(2k+1)!} A + \sum_{k=1}^{\infty} \frac{(-1)^{k-1} \|\omega\|^{2(k-1)}}{(2k)!} A^2 \\ &= I_3 + \left(\sum_{k=0}^{\infty} \frac{(-1)^k \|\omega\|^{2k+1}}{(2k+1)!} \right) \frac{A}{\|\omega\|} + \left(-\frac{1}{\|\omega\|^2} \sum_{k=0}^{\infty} \frac{(-1)^k \|\omega\|^{2k}}{(2k)!} + \frac{1}{\|\omega\|^2} \right) A^2 \\ &= I_3 + \sin \|\omega\| \ \frac{A}{\|\omega\|} + (1 - \cos \|\omega\|) \left(\frac{A}{\|\omega\|} \right)^2 \end{aligned}$$

D'après la formule de Rodrigues, la matrice e^A est une matrice de rotation d'axe dirigé par le vecteur unitaire $\omega/\|\omega\|$ et d'angle $\|\omega\|$.

2) Montrons maintenant que la fonction exponentielle de $\mathfrak{so}(3)$ dans $SO(3)$ est surjective. Prenons d'abord une rotation d'axe (CX) et d'angle a. La matrice de rotation correspondante R_0 s'écrit

$$R_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{pmatrix}.$$

Or,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{pmatrix} = \exp \left(a \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \right) = \exp(a\, L_1).$$

On peut ainsi associer à la rotation R_0 la vitesse angulaire $(\omega_1, \omega_2, \omega_3) = (a, 0, 0)$. Remarquons que la matrice de l'algèbre de Lie obtenue correspond bien à l'approximation au premier ordre de la matrice $R_0 - I_3$. Prenons maintenant une rotation R d'angle a et d'axe quelconque. Par un changement de base adapté, on peut l'écrire sous la forme $R = PR_0P^{-1}$, avec P matrice 3×3 inversible. D'où,

$$R = P \exp \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -a \\ 0 & a & 0 \end{pmatrix} P^{-1} = \exp \left(P \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -a \\ 0 & a & 0 \end{pmatrix} P^{-1} \right).$$

On obtient ainsi les composantes de la vitesse angulaire qui sont les coefficients de la matrice antisymétrique aPL_1P^{-1}.

3) La fonction exponentielle de $\mathfrak{so}(3)$ dans $SO(3)$ n'est pas injective. Par exemple, les vitesses $\omega = (\pi/2, 0, 0)$ et $\omega' = (-3\pi/2, 0, 0)$ sont associées par l'application exponentielle à la même rotation. L'application devient injective si on restreint les valeurs de $\|\omega\|$ à $[0, \pi[$. □

Proposition 1.5 – *Les angles d'Euler représentant une rotation de $SO(3)$ fournissent un élément de l'algèbre de Lie $\mathfrak{so}(3)$ associé à la rotation.*
Si R est une matrice de rotation d'axe dirigé par le vecteur unitaire u et d'angle a, alors les angles d'Euler associés $\omega = au$ vérifient $R = \exp [\omega]_\times$.

Démonstration. On a vu dans la démonstration précédente qu'une matrice de rotation d'axe dirigé par $\omega/\|\omega\|$ et d'angle $\|\omega\|$ était égale à $\exp [\omega]_\times$. Ainsi, pour une rotation d'axe dirigé par le vecteur unitaire u et d'angle a, on peut écrire

$$R = e^{[au]_\times}.$$

Or, nous avons vu dans le paragraphe 1.2.1.3 que le vecteur au est égal au vecteur contenant les angles d'Euler de la rotation. Donc les angles d'Euler associés à R correspondent aux composantes d'une vitesse angulaire de $\mathfrak{so}(3)$ associée à la rotation. □

Notons que la structure d'algèbre est donnée à une algèbre de Lie de matrices par le crochet de Lie, défini pour deux matrices A et B par

$$[A, B] = AB - BA.$$

1.2.2 Modélisation d'un mouvement complet

Définition 1.8 – *Un déplacement rigide de $\mathbb{R}^3$ est une transformation de $\mathbb{R}^3$ dans $\mathbb{R}^3$ qui conserve les angles et les distances.*

L'action d'un déplacement rigide D sur $\mathbb{R}^3$ est composée d'une rotation R suivie d'une translation t

$$\begin{array}{ccc} \mathbb{R}^3 & \longrightarrow & \mathbb{R}^3 \\ x & \longmapsto & Rx + t. \end{array}$$

L'ensemble des déplacements rigides de $\mathbb{R}^3$, notés $D = (R, t)$, forme le groupe Spécial Euclidien $SE(3)$

$$SE(3) = \left\{ (R, t),\ R \in SO(3),\ t \in \mathbb{R}^3 \right\}.$$

Proposition 1.6 – *Le groupe $SE(3)$ est égal au produit semi-direct de $SO(3)$ avec $\mathbb{R}^3$*

$$SE(3) = SO(3) \ltimes \mathbb{R}^3.$$

Démonstration. Ecrivons la composition de deux déplacements $D = D_2 \circ D_1$ avec $D_1 = (R_1, t_1)$ et $D_2 = (R_2, t_2)$

$$x \overset{D_1}{\longmapsto} R_1 x + t_1 \overset{D_2}{\longmapsto} R_2(R_1 x + t_1) + t_2 = R_2 R_1 x + R_2 t_1 + t_2.$$

Donc $D = D_2 \circ D_1 = (R_2 R_1, R_2 t_1 + t_2)$, ce qui implique $SE(3) = SO(3) \ltimes \mathbb{R}^3$. □

Le groupe $SE(3)$ est isomorphe à un sous-groupe de $GL(4, \mathbb{R})$ par l'application

$$\begin{array}{ccc} SE(3) & \longrightarrow & GL(4, \mathbb{R}) \\ (R, t) & \longmapsto & \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}. \end{array}$$

On note aussi par $SE(3)$ l'ensemble de ces matrices

$$SE(3) = \left\{ \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix},\ R \in SO(3),\ t \in \mathbb{R}^3 \right\}.$$

On retrouve la loi de composition $\circ$ correspondant au produit matriciel

$$\begin{pmatrix} R_2 & t_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_1 & t_1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_2 R_1 & R_2 t_1 + t_2 \\ 0 & 1 \end{pmatrix}.$$

Le mouvement d'une caméra est habituellement modélisé dans $SE(3)$; il s'écrit de façon unique comme une rotation R d'axe passant par le centre C de la caméra suivie d'une translation de vecteur t (figure 1.5). On notera dans la suite un déplacement D de caméra par le couple (R, t).

Comme le groupe Spécial Orthogonal, le groupe Spécial Euclidien est un groupe de Lie de matrices (à proprement parler, c'est le sous-groupe de $GL(4, \mathbb{R})$ isomorphe à $SE(3)$ qui est un

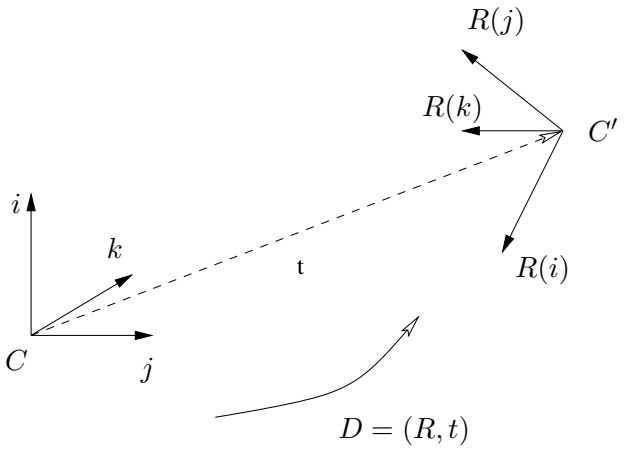

FIGURE 1.5: *Mouvement de caméra.*

groupe de Lie de matrices). Le calcul de l'algèbre de Lie $\mathfrak{se}(3)$ à partir du groupe $SE(3)$ permet d'associer à un déplacement de caméra une vitesse de rotation et une vitesse de translation.

Proposition 1.7 – *L'algèbre de Lie* $\mathfrak{se}(3)$*, associée au groupe de Lie* $SE(3)$*, est égale à*

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} & & & v_1 \\ & [\omega]_\times & & v_2 \\ & & & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ [\omega]_\times \in \mathfrak{so}(3),\ (v_1, v_2, v_3) \in \mathbb{R}^3 \right\}.$$

Remarques

- La matrice $[\omega]_\times$ représente la vitesse angulaire de la caméra et le vecteur (v_1, v_2, v_3) la vitesse de déplacement du centre optique de la caméra.
- On notera aussi
$$\mathfrak{se}(3) = \left\{ (\omega, v),\ \omega \in \mathbb{R}^3,\ v \in \mathbb{R}^3 \right\}.$$

Démonstration. Soit $M \in \mathcal{M}_4(\mathbb{R})$ telle que $\exp(\lambda M) \in SE(3)$ pour tout $\lambda \in \mathbb{R}$. La matrice $\exp(\lambda M)$ doit donc s'écrire sous la forme

$$\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}.$$

Comme $\left.\dfrac{d\, e^{\lambda M}}{d\lambda}\right|_{\lambda=0} = M$, la dernière ligne de la matrice M doit être nulle. Les matrices de $\mathfrak{se}(3)$ sont donc de la forme

$$M = \begin{pmatrix} & y_1 \\ Y & y_2 \\ & y_3 \\ 0 & 0 \end{pmatrix}.$$

On a alors

$$M^k = \begin{pmatrix} Y^k & Y^{k-1}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \\ 0 \quad 0 \end{pmatrix} \text{ donc } e^{\lambda M} = \begin{pmatrix} e^{\lambda Y} & \begin{matrix} * \\ * \\ * \end{matrix} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix}.$$

Pour que $e^{\lambda Y}$ appartienne à $SO(3)$, il faut et il suffit que Y appartienne à l'algèbre de Lie $\mathfrak{so}(3)$, c'est-à-dire que la matrice Y soit antisymétrique.

Ainsi, toute matrice M de la forme $\begin{pmatrix} & & y_1 \\ Y & & y_2 \\ & & y_3 \\ 0 & 0 & 0 \end{pmatrix}$, avec Y appartenant à $\mathfrak{so}(3)$ vérifie la propriété suivante : pour tout réel λ, $\exp(\lambda M)$ appartient à $SE(3)$. □

Proposition 1.8 – *À un couple de vitesses rotationnelle $\omega = (\omega_1, \omega_2, \omega_3)$ et translationnelle $v = (v_1, v_2, v_3)$, l'application exponentielle associe un déplacement de caméra par [64]*

$$\begin{array}{rccc} \exp : & \mathfrak{se}(3) & \longrightarrow & SE(3) \\ & (\omega, v) & \longmapsto & (R, t) \end{array}$$

$$\textit{avec} \begin{cases} R = \exp([\omega]_\times) \\ t = \begin{cases} v & \textit{si } \|\omega\| = 0 \\ \tau(\omega)v = \dfrac{1}{\|\omega\|^2}\left((I-R)[\omega]_\times + \omega\omega^T\right) v & \textit{sinon.} \end{cases} \end{cases} \tag{1.6}$$

L'application exponentielle est surjective et non injective. Elle peut être inversée localement (pour $\|\omega\| \in [0, \pi[$), pour calculer, à partir d'un déplacement (R, t) de caméra, les vitesses ω et v correspondantes.

$$\begin{array}{l} \textit{Si } R = I \quad \textit{alors} \begin{cases} \omega = 0 \\ v = t \end{cases}, \\ \\ \textit{sinon} \quad \begin{cases} \omega \textit{ est calculée à partir de } R \\ v = \tau^{-1}(\omega)t = I_3 - \dfrac{1}{2}[\omega]_\times + \left(\dfrac{1}{\|\omega\|^2} - \dfrac{\sin\|\omega\|}{2\|\omega\|(1-\cos\|\omega\|)}\right)[\omega]_\times^2 \, t. \end{cases} \end{array} \tag{1.7}$$

Démonstration. Quelques éléments sur l'inversion de la matrice $\tau(\omega)$. En utilisant

$$\begin{cases} \omega\omega^T = [\omega]_\times^2 + \|\omega\|^2 I_3 \\ \\ R = e^{[\omega]_\times} = I_3 + \sin\|\omega\| \dfrac{[\omega]_\times}{\|\omega\|} + \dfrac{1-\cos\|\omega\|}{\|\omega\|^2}[\omega]_\times^2 \\ \\ [\omega]_\times^3 = -\|\omega\|^2[\omega]_\times, \end{cases}$$

on peut réécrire $\tau(\omega)$ de la façon suivante

$$\tau(\omega) = I_3 + \frac{1-\cos\|\omega\|}{\|\omega\|^2}[\omega]_\times + \frac{\|\omega\| - \sin\|\omega\|}{\|\omega\|^3}[\omega]_\times^2.$$

Le déterminant de $\tau(\omega)$ vaut $\det(\tau(\omega)) = \dfrac{2}{\|\omega\|^2}(1-\cos\|\omega\|)$, donc $\tau(\omega)$ est inversible pour $\|\omega\| \in]0, \pi[$. Par la méthode des cofacteurs, on inverse la matrice $\tau(\omega)$ et on obtient

$$\tau^{-1}(\omega) = I_3 - \frac{1}{2}[\omega]_\times + \left(\frac{1}{\|\omega\|^2} - \frac{\sin\|\omega\|}{2\|\omega\|(1-\cos\|\omega\|)}\right)[\omega]_\times^2.$$

□

1.3 Flot optique généré par un mouvement de caméra

1.3.1 Définition et évaluation du flot optique

Le terme de flot optique a été inventé par le psychologue James Jerome Gibson en 1950 dans une étude sur la vision humaine [21]. Le flot optique entre deux images successives est le mouvement apparent des pixels d'une image à l'autre. C'est un champ de vecteurs représentant les déplacements des points de la première image à la seconde. Gibson conjecture qu'il y a suffisamment d'information dans le flot optique pour déduire une unique interprétation physiquement correcte du mouvement tridimensionnel (à un facteur d'échelle près pour la translation) et de la structure de la scène. C'est généralement le cas sauf pour un ensemble de surfaces filmées de mesure nulle. Dans le chapitre 5, on discutera l'injectivité de la fonction associant à un mouvement de caméra et une surface filmée le flot optique correspondant, suivant les domaines d'observation du flot.

Pour estimer le flot optique sur une séquence d'images, on fait l'hypothèse d'éclairement constant au cours du temps. Si (x, y) sont les coordonnées d'un point suivi dans la séquence d'images, on note $(x(t), y(t), t)$ la trajectoire 2D du point au cours du temps dans la séquence. Si $I(x(t), y(t), t)$ est l'intensité lumineuse au point $(x(t), y(t))$ sur l'image au temps t, l'hypothèse d'éclairement constant s'écrit

$$I(x(t), y(t), t) = \text{constante}.$$

En dérivant par rapport au temps cette expression, on obtient la contrainte du flot optique

$$I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t = 0 \tag{1.8}$$

où I_x, I_y, I_t sont les dérivées partielles de l'image $I(x, y, t)$. Le vecteur $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ que nous noterons dans la suite $u(x, y, t)$, représente la vitesse du point (x, y) sur le plan rétinien au temps t. Cette quantité est appelée flot optique.

Cependant, étant donnée une séquence d'images, l'équation (1.8) est insuffisante à la détermination d'un flot optique unique. En effet, le problème est mal posé car le flot optique a deux composantes et nous ne disposons que d'une seule équation. Ce problème est appelé problème d'ouverture. Pour le surmonter, il est nécessaire de poser une seconde hypothèse ; c'est souvent une contrainte de régularité ou de lissage du flot. En 1980, Horn et Schunck [32] proposent une méthode basée sur la régularisation ; à la contrainte du flot optique est ajoutée une contrainte de régularité spatiale du flot. La fonctionnelle totale à minimiser est la suivante

$$\int_\Omega \left[\left(\begin{pmatrix} I_x \\ I_y \end{pmatrix} .u(x, y, t) + I_t \right)^2 + \lambda \left\| \triangle u(x, y, t) \right\|^2 \right] dx\, dy$$

où Ω est le domaine spatial considéré. Le paramètre λ détermine l'importance accordée à la régularisation. Le travail de Horn et Schunck a été suivi par un grand nombre de contributions pour le calcul du flot optique, utilisant différentes méthodes : les méthodes de corrélations locales (ou block-matching) consistant à comparer de petites zones locales d'une image avec les zones voisines de l'image suivante, les méthodes basées sur le gradient des images, les méthodes de filtrage spatio-temporel... Un inventaire des méthodes existantes en 1994 a été réalisé par Baron, Fleet et Beauchemin [3].

La méthode numérique que nous avons choisie pour évaluer un flot optique sur une séquence, pour les expériences évoquées dans le chapitre 3, est celle de Weickert et Schnörr [74]. Leur approche consiste à ajouter à la contrainte du flot optique (1.8) l'hypothèse que le flot admet une régularité spatiale mais aussi temporelle ; ceci permet de considérer non seulement les modifications spatiales d'une image à la suivante mais aussi globalement dans toute la séquence. La fonctionnelle à minimiser, sur un domaine spatio-temporel $\Omega \times [0, T]$ de la séquence, est la suivante

$$E(u(x, y, t)) = \int_{\Omega \times [0,T]} \left[\left(\begin{pmatrix} I_x \\ I_y \end{pmatrix} .u(x, y, t) + I_t \right)^2 + \lambda\, \psi \left(\left\| \nabla u(x, y, t) \right\|^2 \right) \right] dx\, dy\, dt$$

où $\psi(s^2) = \epsilon s^2 + (1 - \epsilon)\beta^2 \sqrt{1 + \frac{s^2}{\beta^2}}$ et l'opérateur ∇ est l'opérateur de gradient spatio-temporel $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t}\right)$. Le poids λ, appelé paramètre de régularisation, détermine l'importance accordée au lissage. Weickert et Schnörr ont montré que la contrainte de régularité spatio-temporelle

du flot optique permet de réduire les effets du bruit par rapport à une régularité uniquement spatiale.

Remarque – On appelle flot optique en un point (x, y) du plan rétinien, la vitesse de ce point $u(x, y, t) = (dx(t)/dt,\ dy(t)/dt)$ sur le plan. Par extension, on appellera aussi flot optique, le déplacement de (x, y) à (x', y') entre deux instants, c'est-à-dire $(x' - x, y' - y)$.

1.3.2 Relation entre vitesse de la caméra et flot optique

On s'intéresse ici aux liens entre la vitesse d'une caméra et le flot optique engendré entre les images.

Proposition 1.9 – *Soit une caméra en mouvement ayant une vitesse de translation $v(t) = (v_1(t), v_2(t), v_3(t))$ et de rotation $\omega(t) = (\omega_1(t), \omega_2(t), \omega_3(t))$. Alors le flot optique $u(x, y, t)$ en un point (x, y) du plan rétinien vérifie*

$$\begin{aligned} u(x,y,t) &= \frac{1}{Z}\begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix} v(t) + \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{pmatrix} \omega(t) \\ &= \frac{1}{Z}\, A(x,y)\, v(t) + B(x,y)\, \omega(t) \end{aligned} \tag{1.9}$$

où Z est la profondeur du point de l'espace projeté au point (x, y) dans le repère associé au plan rétinien.

Démonstration. Considérons une caméra en mouvement ayant une vitesse de translation $v(t)$ et de rotation $\omega(t)$ et un point M de l'espace de coordonnées (X, Y, Z) dans le repère (C, i, j, k) de la caméra. Cela revient à considérer que le point M se déplace suivant une trajectoire $M(t)$ dans le repère $3D$ de la caméra à une vitesse $dM(t)/dt = (dX(t)/dt,\ dY(t)/dt,\ dZ(t)/dt)$

$$\frac{dM(t)}{dt} = -v(t) - [\omega(t)]_\times (X(t), Y(t), Z(t))^T. \tag{1.10}$$

Soit $m(t)$ la projection du point $M(t)$ sur le plan rétinien $\mathcal{R}$. Ses coordonnées $(x(t), y(t))$ dans le repère (c, i, j) de $\mathcal{R}$ vérifient

$$\begin{cases} x(t) = \dfrac{X(t)}{Z(t)} \\ \\ y(t) = \dfrac{Y(t)}{Z(t)}. \end{cases}$$

En dérivant les trajectoires $x(t)$ et $y(t)$ par rapport au temps, on obtient le flot optique $u(x, y, t) =$

$(dx(t)/dt,\ dy(t)/dt)$

$$\begin{cases} \dfrac{dx(t)}{dt} = \dfrac{\dot{X}(t)Z(t) - X(t)\dot{Z}(t)}{Z(t)^2} \\ \\ \dfrac{dy(t)}{dt} = \dfrac{\dot{Y}(t)Z(t) - Y(t)\dot{Z}(t)}{Z(t)^2}. \end{cases}$$

En utilisant la relation (1.10), $dX(t)/dt$, $dY(t)/dt$ et $dZ(t)/dt$ s'écrivent en fonction des vitesses translationnelle $v(t)$ et rotationnelle $\omega(t)$ de la caméra. Pour alléger les formules, on omet dans ce qui suit la variable temporelle

$$\begin{cases} \dfrac{dX}{dt} = -v_1 + \omega_3 Y - \omega_2 Z = -v_1 + Z(\omega_3 y - \omega_2) \\ \\ \dfrac{dY}{dt} = -v_2 - \omega_3 X + \omega_1 Z = -v_2 + Z(-\omega_3 x + \omega_1) \\ \\ \dfrac{dZ}{dt} = -v_3 + \omega_2 X - \omega_1 Y = -v_3 + Z(\omega_2 x - \omega_1 y). \end{cases}$$

En remplaçant dX/dt, dY/dt et dZ/dt dans les expressions de dx/dt et dy/dt, on obtient

$$\begin{cases} \dfrac{dx}{dt} &= \dfrac{(-v_1 + Z(\omega_3 y - \omega_2))Z - Zx(-v_3 + Z(\omega_2 x - \omega_1 y))}{Z^2} \\ \\ &= \dfrac{-v_1}{Z} + \omega_3 y - \omega_2 + \dfrac{v_3 x}{Z} - x(\omega_2 x - \omega_1 y) \\ \\ \dfrac{dy}{dt} &= \dfrac{(-v_2 + Z(-\omega_3 x + \omega_1))Z - Zy(-v_3 + Z(\omega_2 x - \omega_1 y))}{Z^2} \\ \\ &= \dfrac{-v_2}{Z} - \omega_3 x + \omega_1 + \dfrac{v_3 y}{Z} - y(\omega_2 x - \omega_1 y) \end{cases}$$

soit une formule linéaire en (ω, v) donnant le flot optique $(dx(t)/dt,\ dy(t)/dt) = u(x, y, t)$ en un point (x, y) en fonction des vitesses v et ω et de la profondeur Z du point projeté

$$u(x, y, t) = \frac{1}{Z} \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix} v(t) + \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{pmatrix} \omega(t).$$

□

Le déplacement d'un point d'une image à l'autre dépend donc de la profondeur du point de l'espace projeté. Si la caméra effectue une translation, plus le point projeté est éloigné de la caméra, plus le déplacement observé sur l'image et dû à la vitesse $v(t)$ est petit.

1.4 Relations entre deux vues d'une scène fixe

Dans la section précédente, nous avons modélisé le mouvement d'une caméra dans l'espace. Dans cette section, nous nous intéressons aux images produites par la caméra avant et après le déplacement, lorsque la scène filmée est fixe, et en supposant l'éclairement constant. Est-il alors possible d'associer les points des images deux à deux ? Dans le cas d'un mouvement de caméra quelconque, ce n'est pas possible à cause de l'effet de parallaxe. Cependant, dans le cas où les positions du centre optique lors des deux prises de vue sont différentes (c'est-à-dire quand la caméra s'est translatée), il existe une contrainte géométrique importante liant des points appariés de deux images, c'est-à-dire représentant le même point 3D. C'est la contrainte épipolaire. Nous allons d'abord présenter l'effet de parallaxe, puis les deux cas particuliers pour lesquels il est possible d'associer les points deux à deux, enfin nous expliciterons la contrainte épipolaire.

1.4.1 Effet de parallaxe

Considérons une caméra filmant une scène fixe soumise à un déplacement D. On acquiert ainsi deux vues différentes de la scène. Deux points, l'un de la première image, l'autre de la seconde, sont dits appariés (ou en correspondance) s'ils sont les projections d'un même point de l'espace 3D. En général, on ne peut associer des points de la première image à des points de la seconde car la visibilité d'un point de l'espace dépend de sa profondeur. C'est l'effet de parallaxe illustré sur la figure (1.6). Les points M_1 et M_2 de l'espace ont même projection sur la première image et deux projections distinctes sur la deuxième image, du fait de leurs profondeurs différentes.

Deux cas particuliers ne sont pas concernés par le problème de l'effet de parallaxe, ce qui rend alors possible la mise en correspondance des points des images.

1.4.2 Cas d'une scène plane filmée

Considérons une caméra de longueur focale égale à 1. Notons C le centre optique de la caméra, $\mathcal{R}$ le plan rétinien associé. Soit $D = (R, t)$ un déplacement de la caméra, C' la nouvelle position du centre optique, et $\mathcal{R}'$ le nouveau plan rétinien (après le déplacement). Lorsque la caméra filme un plan Π de l'espace ne passant pas par les centres optiques C et C', il n'y a pas d'effet de parallaxe car aucun phénomène d'occultation d'une image à l'autre ne peut se produire. Il existe alors une homographie liant les images produites sur les deux plans $\mathcal{R}$ et $\mathcal{R}'$, c'est-à-dire une transformation de $\mathcal{R}$ dans $\mathcal{R}'$ linéaire en coordonnées projectives [15]. Nous allons expliciter cette relation en coordonnées euclidiennes.

Soient (C, i, j, k) et $(C', R(i), R(j), R(k))$ les systèmes standards normalisés associés à la caméra respectivement avant et après son déplacement. Les plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ sont munis des repères (c, i, j) et $(c', R(i), R(j))$ où c et c' sont les points principaux associés à C et C'.

Proposition 1.10 – *Soit une caméra filmant un plan* Π *et ayant un mouvement* $D = (R, t)$.

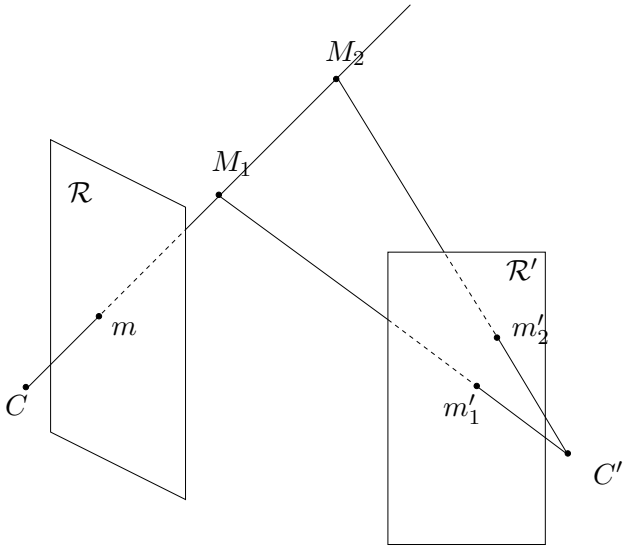

FIGURE 1.6: *L'effet de parallaxe.*

Soient f et g les images des plans $\mathcal{R}$ et $\mathcal{R}'$ formées avant et après le déplacement. On note

$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \text{ et } t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}.$$

Alors, à tout point M du plan Π, sont associées deux projections m et m' sur les plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ vérifiant

$$f(m) = g(m').$$

Si on note (x, y) les coordonnées de m dans (c, i, j) et (x', y') celles de m' dans $(c', R(i), R(j))$ alors

$$\begin{cases} x' = \dfrac{a_1 x + a_2 y + a_3 - \dfrac{a_1 t_1 + a_2 t_2 + a_3 t_3}{Z}}{c_1 x + c_2 y + c_3 - \dfrac{c_1 t_1 + c_2 t_2 + c_3 t_3}{Z}} \\ y' = \dfrac{b_1 x + b_2 y + b_3 - \dfrac{b_1 t_1 + b_2 t_2 + b_3 t_3}{Z}}{c_1 x + c_2 y + c_3 - \dfrac{c_1 t_1 + c_2 t_2 + c_3 t_3}{Z}} \end{cases} \tag{1.11}$$

et

$$\begin{cases} x = \dfrac{a_1 x' + b_1 y' + c_1 + \dfrac{t_1}{Z'}}{a_3 x' + b_3 y' + c_3 + \dfrac{t_3}{Z'}} \\ \\ y = \dfrac{a_2 x' + b_2 y' + c_2 + \dfrac{t_2}{Z'}}{a_3 x' + b_3 y' + c_3 + \dfrac{t_3}{Z'}}, \end{cases} \tag{1.12}$$

où Z est la profondeur du point M dans le repère (C, i, j, k) et Z' est la profondeur de M dans le repère $(C', R(i), R(j), R(k))$.

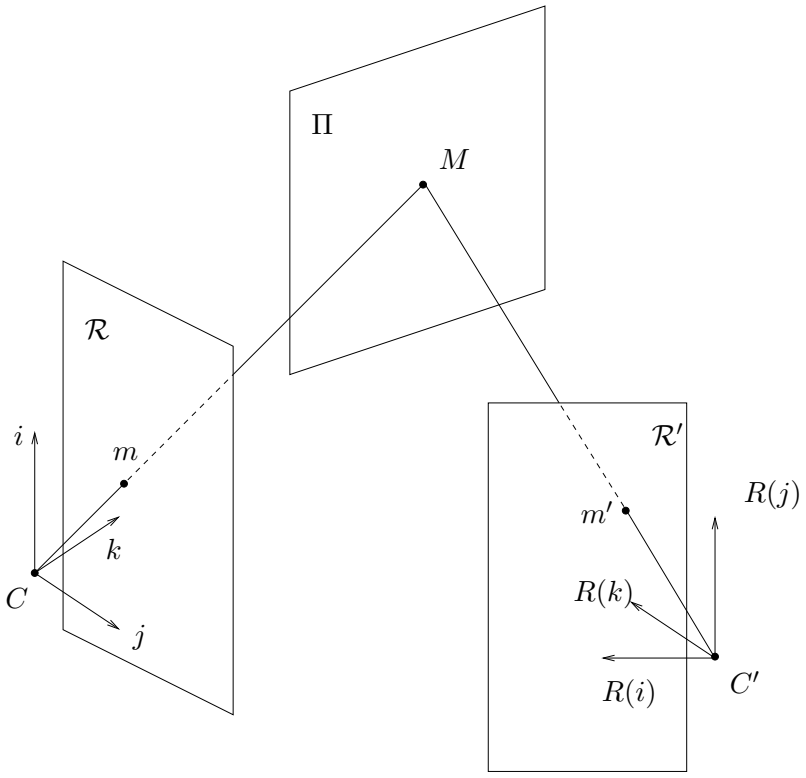

FIGURE 1.7: *Cas d'une scène plane filmée : absence d'effet de parallaxe.*

Démonstration. Soit M un point du plan Π, m sa projection sur $\mathcal{R}$ et m' sa projection sur $\mathcal{R}'$. Comme l'illumination de la scène est supposée constante en espace et en temps, les niveaux de gris en m et m' sont les mêmes sur les deux images soit

$$f(m) = g(m').$$

Soit (X, Y, Z) les coordonnées du point M dans le repère (C, i, j, k) et (X', Y', Z') ses coordonnées dans $(C', R(i), R(j), R(k))$. Les coordonnées du point m dans le repère (c, i, j) sont donc, d'après

la formule (1.1),

$$\begin{cases} x = \dfrac{X}{Z} \\ \\ y = \dfrac{Y}{Z} \end{cases}$$

et celles du point m' dans le repère $(c', R(i), R(j))$

$$\begin{cases} x' = \dfrac{X'}{Z'} \\ \\ y' = \dfrac{Y'}{Z'}. \end{cases}$$

Exprimons les coordonnées (x', y') en fonction des coordonnés (x, y) ; nous avons

$$\overrightarrow{C'M} = \overrightarrow{C'C} + \overrightarrow{CM}$$

$$\Leftrightarrow X'R(i) + Y'R(j) + Z'R(k) = -(t_1 i + t_2 j + t_3 k) + (Xi + Yj + Zk)$$

$$\Leftrightarrow X' \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + Y' \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + Z' \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = - \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\Leftrightarrow R \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = - \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = -R^{-1} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + R^{-1} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1.13}$$

d'où

$$\begin{cases} x' = \dfrac{X'}{Z'} = \dfrac{a_1 X + a_2 Y + a_3 Z - (a_1 t_1 + a_2 t_2 + a_3 t_3)}{c_1 X + c_2 Y + c_3 Z - (c_1 t_1 + c_2 t_2 + c_3 t_3)} \\ \\ y' = \dfrac{Y'}{Z'} = \dfrac{b_1 X + b_2 Y + b_3 Z - (b_1 t_1 + b_2 t_2 + b_3 t_3)}{c_1 X + c_2 Y + c_3 Z - (c_1 t_1 + c_2 t_2 + c_3 t_3)} \end{cases}$$

$\Leftrightarrow$

$$\begin{cases} x' = \dfrac{a_1 x + a_2 y + a_3 - \dfrac{a_1 t_1 + a_2 t_2 + a_3 t_3}{Z}}{c_1 x + c_2 y + c_3 - \dfrac{c_1 t_1 + c_2 t_2 + c_3 t_3}{Z}} \\ \\ y' = \dfrac{b_1 x + b_2 y + b_3 - \dfrac{b_1 t_1 + b_2 t_2 + b_3 t_3}{Z}}{c_1 x + c_2 y + c_3 - \dfrac{c_1 t_1 + c_2 t_2 + c_3 t_3}{Z}} \end{cases}$$

donc il existe une application liant les points de la première image f à ceux de la deuxième image g, connaissant les profondeurs des points du plan filmé. On peut aussi exprimer (x, y) en fonction de (x', y') par un calcul analogue

$$\begin{cases} x = \dfrac{X}{Z} = \dfrac{a_1X' + b_1Y' + c_1Z' + t_1}{a_3X' + b_3Y' + c_3Z' + t_3} \\ \\ y = \dfrac{Y}{Z} = \dfrac{a_2X' + b_2Y' + c_2Z' + t_2}{a_3X' + b_3Y' + c_3Z' + t_3} \end{cases}$$

$\Leftrightarrow$

$$\begin{cases} x = \dfrac{a_1x' + b_1y' + c_1 + \dfrac{t_1}{Z'}}{a_3x' + b_3y' + c_3 + \dfrac{t_3}{Z'}} \\ \\ y = \dfrac{a_2x' + b_2y' + c_2 + \dfrac{t_2}{Z'}}{a_3x' + b_3y' + c_3 + \dfrac{t_3}{Z'}}. \end{cases}$$

Par conséquent, lorsque l'on connaît les profondeurs des points du plan Π (ou l'équation du plan), on peut déterminer les correspondances des points entre les images f et g. $\square$

1.4.3 Cas d'une rotation de caméra

Le deuxième cas particulier dans lequel l'effet de parallaxe ne survient pas, concerne non plus l'objet du film mais un mouvement particulier de caméra. En effet, le phénomène de parallaxe est dû à la translation de la caméra dans l'espace. Si les centres optiques de la caméra avant et après son mouvement sont confondus, un point de l'espace occulté lors de la projection sur le premier plan rétinien ne pourra pas apparaître sur le second. La figure (1.8) illustre ce phénomène. Tous les points d'une droite passant par le centre optique ont la même projection après une rotation de la caméra tandis qu'ils ont des projections différentes dans le cas d'une translation.

En reprenant les notations de la partie précédente, et dans le cas d'un mouvement de caméra constitué d'une rotation seule, un point de coordonnées (x, y) de l'image f sera apparié à un point de coordonnées (x', y') de l'image g par l'application

$$\begin{cases} x' = \dfrac{a_1x + a_2y + a_3}{c_1x + c_2y + c_3} \\ \\ y' = \dfrac{b_1x + b_2y + b_3}{c_1x + c_2y + c_3} \end{cases}$$

et réciproquement un point (x', y') de l'image g correspondra à un point (x, y) de l'image f par

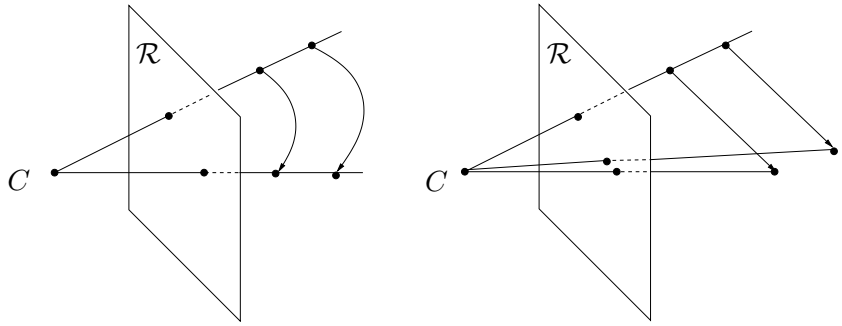

FIGURE 1.8: *A gauche, la caméra effectue une rotation (ce qui revient à considérer que la caméra est fixe et que les points de l'espace sont en rotation autour d'un axe passant par le centre optique C) ; les points d'une droite passant par C sont projetés, après la rotation, en un seul point sur le plan rétinien. À droite, la caméra se translate (ce qui revient à translater les points de l'espace) ; les points d'une droite passant par C sont projetés, après la translation, en plusieurs points.*

l'application

$$\begin{cases} x = \dfrac{a_1x' + b_1y' + c_1}{a_3x' + b_3y' + c_3} \\ \\ y = \dfrac{a_2x' + b_2y' + c_2}{a_3x' + b_3y' + c_3}. \end{cases}$$

L'effet de parallaxe ne peut être observé car les déformations sont indépendantes des profondeurs des objets filmés.

1.4.4 La contrainte épipolaire

Plaçons-nous maintenant dans le cas général d'un mouvement complet de caméra et d'une scène filmée 3D. Soit $D = (R, t)$ un déplacement de la caméra. On considère ici le cas où la translation est non nulle, le cas d'un mouvement sans translation ayant été traité dans le paragraphe précédent. Soit M un point de l'espace et ses images m et m' sur les images f et g des plans $\mathcal{R}$ et $\mathcal{R}'$. Connaissant le mouvement de la caméra et les deux vues correspondantes, on ne peut mettre en correspondance le point m de f avec le point m' de g à cause de l'effet de parallaxe. Cependant, le point m' ne peut se placer en tous les points de g : c'est la contrainte épipolaire énoncée ci-après.

En utilisant les mêmes notations que dans la partie 1.4.2, la ligne CC' intersecte les plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ respectivement en deux points e et e' appelés épipoles. Les droites de $\mathcal{R}$ et $\mathcal{R}'$ passant par e et e' sont appelées droites ou lignes épipolaires. Dans le cas où le plan $\mathcal{R}$ ou le plan $\mathcal{R}'$ est parallèle à la droite CC', l'épipole contenu dans ce plan est à l'infini et les droites épipolaires de ce plan sont des droites parallèles à la droite CC'.

Le rayon optique (CM) ou (Cm) est projeté sur l'image g en une droite passant par l'épipole e' et par l'image d'un point du rayon. Cette droite est appelée l'_m ligne épipolaire de m', car le

point m' doit appartenir à cette droite. De la même façon, pour tout point m' appartenant à g, le point correspondant dans f, m, doit appartenir à la ligne épipolaire $l_{m'}$. C'est la contrainte épipolaire, illustrée sur la figure (1.9).

Les paragraphes suivants développent les notions de matrice fondamentale et matrice essentielle, notions projectives développées par Faugeras [13, 15] et Hartley [26] liant les coordonnées projectives d'un point m à la ligne épipolaire correspondante.

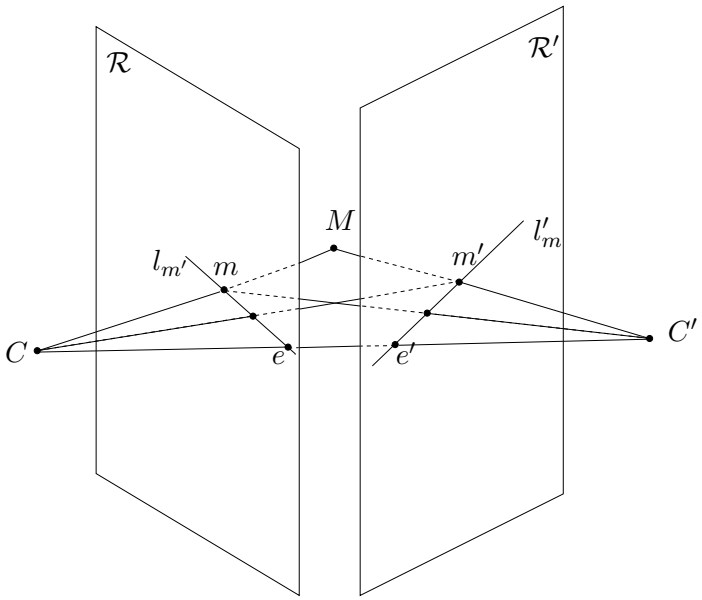

FIGURE 1.9: *Illustration de la contrainte épipolaire.*

1.4.4.1 La matrice fondamentale

Intéressons-nous d'abord à la représentation d'une droite en coordonnées projectives. Soit une droite l d'un plan de $\mathbb{R}^3$. Dans le plan, elle a pour équation $au + bv + c = 0$. Soit maintenant un point du plan appartenant à la droite l de coordonnées projectives (x, y, z) soit de coordonnées euclidiennes $(x/z, y/z)$. Ces dernières vérifient

$$a\,\frac{x}{z} + b\,\frac{y}{z} + c = 0 \Leftrightarrow ax + by + cz = 0$$

soit

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}^T \begin{pmatrix} x \\ y \\ z \end{pmatrix} = 0.$$

On appelle représentation projective de la droite l, définie à un scalaire près, le point 2D de coordonnées projectives $\mathbf{l}=(a, b, c)$. Les points et les droites sont donc représentés par des triplets

en coordonnées projectives. Si un point m appartient à la droite l, en réécrivant l'expression précédente, on a

$$\mathbf{l^T m} = 0.$$

On peut déduire de la linéarité en coordonnées projectives du modèle de caméra sténopé que la relation entre les coordonnées projectives $\mathbf{m}$ d'un point m du plan rétinien $\mathcal{R}$ et les coordonnées projectives $\mathbf{l_m}$ de sa ligne épipolaire dans $\mathcal{R}'$ est aussi une application linéaire. Ceci provient du fait que les relations entre un point $\mathbf{m}$ et le rayon optique (CM) et entre (CM) et sa projection en coordonnées projectives dans $\mathcal{R}'$ sont linéaires.

La matrice fondamentale F décrit l'application linéaire en coordonnées projectives, liant un point m de $\mathcal{R}$ à sa ligne épipolaire l'_m dans $\mathcal{R}'$. Elle traduit algébriquement la contrainte épipolaire

$$\mathbf{l'_m} = F\mathbf{m}.$$

Proposition 1.11 – *Si F est la matrice fondamentale et m et m' sont deux points appariés de $\mathcal{R}$ et $\mathcal{R}'$, alors la contrainte épipolaire est exprimée par*

$$\mathbf{m'}^T F\mathbf{m} = \mathbf{m}^T F^T \mathbf{m'} = 0. \tag{1.14}$$

Démonstration. Soit un point m du plan $\mathcal{R}$ exprimé en coordonnées projectives et m' le point correspondant du plan $\mathcal{R}'$. La représentation projective $\mathbf{l'_m}$ de la ligne épipolaire de m vérifie

$$\mathbf{l'_m} = F\mathbf{m},$$

ce qui signifie aussi, comme $m' \in l'_m$,

$$\mathbf{m'}^T F\mathbf{m} = 0 \Leftrightarrow \mathbf{m}^T F^T \mathbf{m'} = 0.$$

Donc si la ligne épipolaire $\mathbf{l'_m}$ de m est représentée par $F\mathbf{m}$, la ligne épipolaire $\mathbf{l_{m'}}$ de m' est représentée par $F^T\mathbf{m'}$. La contrainte épipolaire est exprimée algébriquement par

$$\mathbf{m'}^T F\mathbf{m} = \mathbf{m}^T F^T \mathbf{m'} = 0.$$

□

Exprimons maintenant la matrice F en fonction de la matrice de projection de la caméra et de la position des épipoles sur les images f et g.

Proposition 1.12 – *Soient $\mathcal{P}$ et $\mathcal{P}'$ les matrices de projection associées à la caméra respectivement avant et après son déplacement, et $\mathcal{P}^+$ une matrice de projection inverse de $\mathcal{P}$*

(c'est-à-dire vérifiant $\mathcal{P}\mathcal{P}^+ = \lambda I_3$ pour une valeur λ réelle). Alors la matrice fondamentale F s'écrit en fonction de $\mathcal{P}^+$, $\mathcal{P}'$ et de l'épipole e'

$$F = [\mathbf{e}']_\times \mathcal{P}'\mathcal{P}^+.$$

Cette proposition est démontrée dans [15]. La matrice fondamentale F associée à une caméra et à son déplacement n'est donc pas unique puisqu'elle s'écrit en fonction d'une matrice de projection inverse non unique. C'est une matrice de format 3×3 et de rang 2 (car la matrice $[\mathbf{e}']_\times$ est également de rang 2), à 7 degrés de liberté (car c'est une matrice homogène de déterminant nul). Les épipoles e et e' vérifient

$$F\mathbf{e} = 0 \quad \text{et} \quad F^T\mathbf{e}' = 0.$$

1.4.4.2 La matrice essentielle

Supposons maintenant la caméra calibrée, c'est-à-dire la matrice A contenant les paramètres intrinsèques connue, alors la matrice de projection de la caméra est $\mathcal{P} = A\mathcal{P}_0 = A[I_3|0]$, d'après la formule (1.4). Soit un point m du plan rétinien $\mathcal{R}$, image d'un point M de l'espace. On appelle coordonnées normalisées de m

$$\widehat{\mathbf{m}} = A^{-1}\mathbf{m}.$$

Alors, d'après l'équation (1.14),

$$\mathbf{m}'^T F\mathbf{m} = (A\widehat{\mathbf{m}'})^T F(A\widehat{\mathbf{m}}) = \widehat{\mathbf{m}'}^T(A^T F A)\widehat{\mathbf{m}} = 0.$$

La matrice $A^T F A$ est appelée matrice essentielle E. L'équation liant les coordonnées normalisées de deux points de $\mathcal{R}$ et $\mathcal{R}'$

$$\widehat{\mathbf{m}'}^T E\widehat{\mathbf{m}} = 0 \tag{1.15}$$

est connue sous le nom d'équation de Longuet-Higgins [42]. Dans [13], il est montré que

$$E = [t]_\times R.$$

Géométriquement, cette contrainte traduit la coplanarité des vecteurs $\overrightarrow{Cm}$, $\overrightarrow{C'm'}$ et t (soit $\overrightarrow{CC'}$). Historiquement introduite avant la matrice fondamentale, la matrice essentielle a seulement 5 degrés de liberté : 3 pour la rotation, 3 pour la translation moins un car l'échelle de la translation est indéterminée.

1.4.4.3 Différences entre les matrices fondamentale et essentielle

La matrice essentielle E peut être décomposée en un vecteur translation et une matrice de rotation. Dans l'équation dans laquelle elle intervient $\widehat{\mathbf{m}'}^T E\widehat{\mathbf{m}} = 0$, m et m' sont exprimés en coordonnées projectives dans le repère de la caméra respectivement avant et après le déplacement. Elle est donc utile pour l'analyse de mouvement dans le cas d'une caméra déjà calibrée.

La matrice fondamentale F s'écrit en fonction des paramètres de l'homographie épipolaire. Dans l'équation $\mathbf{m'}^T F\mathbf{m} = 0$, m et m' sont exprimés en coordonnées projectives dans le repère de référence de l'espace $\mathbb{R}^3$; F est utilisée dans le cas d'une caméra non calibrée.

Ces deux matrices expriment la même contrainte épipolaire dans deux systèmes de coordonnées différents et peuvent toutes deux se décomposer en produit d'une matrice antisymétrique et d'une autre matrice. Dans le cas de E, cette autre matrice est la matrice de rotation correspondant au déplacement.

1.5 État de l'art de l'estimation d'un mouvement de caméra

À ce niveau du chapitre, nous avons présenté le modèle de caméra projectif, modélisé le mouvement d'une caméra, explicité le flot optique et son calcul, et résumé les relations entre deux vues d'une scène fixe. Tous ces éléments étant posés, nous pouvons présenter le problème étudié dans ce document : l'estimation du mouvement d'une caméra filmant une scène fixe à partir de la séquence d'images produites.

Les méthodes existantes pour résoudre ce problème sont très diverses. La difficulté réside dans le fait que le mouvement d'un pixel entre deux images dépend non seulement des paramètres du mouvement de la caméra mais aussi de la profondeur du point projeté. La figure (1.10) illustre la difficulté à estimer un mouvement de caméra à partir d'un flot optique à cause des différentes profondeurs de la scène filmée.

La revue que nous présentons ci-après est loin d'être exhaustive, étant donné le grand nombre de publications sur le sujet durant les vingt-cinq dernières années. Nous évoquerons quelques approches dans ce qui suit et nous les séparerons, classiquement, en trois catégories : les méthodes directes, les méthodes discrètes ou en temps discret et les méthodes différentielles ou en temps instantané.

Les méthodes directes utilisent directement l'information fournie par le contenu des images, sans suivi de points préalable, ni calcul de flot optique.

Les méthodes discrètes utilisent des correspondances de points entre les images, elles appliquent donc des techniques de suivi (ou tracking) de points singuliers ou de marqueurs dans une séquence ; lorsqu'un nombre de points supérieur à 5 est suivi, le problème d'estimation du mouvement est surdéterminé. À la différence des méthodes directes, elles minimisent une mesure d'erreur sur seulement quelques points et non sur tous les pixels.

Les méthodes différentielles utilisent le flot optique, champ des vitesses des points de l'image, et calculent ainsi le mouvement 3D de la caméra et souvent les profondeurs des objets filmés. Remarquons que la mise en correspondance de points entre deux images est aussi un moyen de mesurer le flot optique ; le suivi de points peut donc aussi bien être exploité par des algorithmes en temps discret qu'en temps instantané. Les méthodes différentielles sont en fait une extrapolation des méthodes discrètes, extrapolation valide pour des pas de temps courts entre deux captures d'images. Ces deux types d'approches dépendent de la précision de la détection, qui n'est pas toujours assurée.

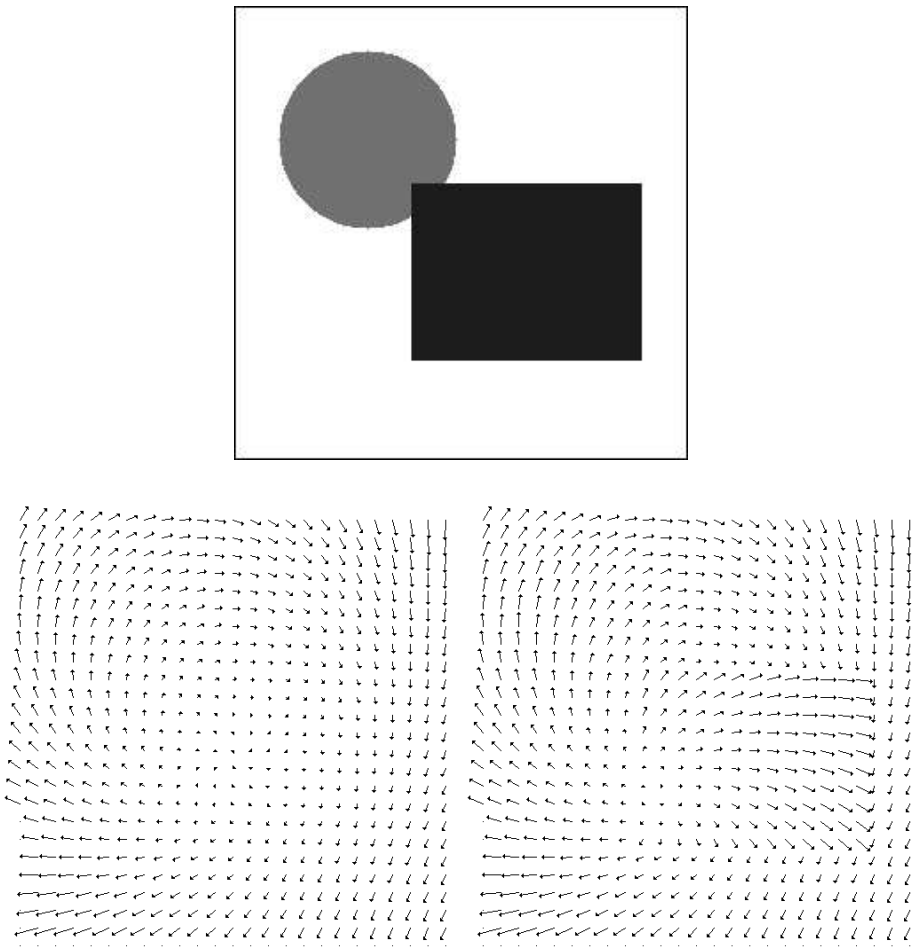

FIGURE 1.10: *En haut, la scène utilisée pour générer le flot optique présenté au-dessous à droite. Le rectangle, le disque et le fond ont respectivement pour profondeur* 2, 5 *et* 10 *dans le repère de la caméra avant le déplacement. En bas, flots optiques générés par une caméra en mouvement, de longueur focale égale à* 1, *filmant à gauche, une scène quelconque de profondeur uniforme égale à* 10 *dans le repère de la caméra, et à droite la scène présentée ci-dessus. Le mouvement de la caméra est une rotation d'axe* $(0.01, 0.01, -1)$, *d'angle* 1.7 *degrés suivie d'une translation de vecteur* $(3, -5, 0.1)$ *(en pixels). Au niveau des discontinuités de profondeurs, les discontinuités du champ de flot sont visibles.*

D'autres critères distinguent aussi les méthodes d'estimation ; l'estimation ou non de la structure de la scène (après calcul du mouvement ou simultanément), l'ordre d'estimation de la rotation et de la translation, l'utilisation de techniques d'optimisation numérique (moindres carrés), ou de techniques incrémentales. Un état de l'art succinct est présenté ci-après.

1.5.1 Méthodes directes

Les méthodes qui n'utilisent ni flot optique, ni points appariés, mais seulement le contenu d'une paire d'images pour estimer le mouvement de la caméra, sont appelées méthodes directes. Irani et Anandan décrivent dans [35] leur principe général. L'équation de base sur laquelle elles reposent est la contrainte d'illumination constante (1.8), utilisée par les algorithmes de calcul du flot optique. Entre deux images données f et g, la contrainte s'écrit

$$f(x,y) = g\left((x,y) + u(x,y,t)\ dt\right)$$

où $u(x,y,t)$ est le flot optique au point (x,y), c'est-à-dire le déplacement du pixel (x,y) entraîné par le mouvement de la caméra et dt est l'intervalle de temps entre les acquisitions des images f et g.

À cette contrainte, s'ajoute un modèle de mouvement explicitement choisi, rotation ou translation pure par exemple. Dans le cas d'un mouvement instantané, entre deux images consécutives d'une séquence par exemple, le modèle est donné par l'équation (1.9)

$$u(x,y,t) = \frac{1}{Z}\begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix} v(t) + \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{pmatrix} \omega(t)$$

et on cherche à estimer les vitesses $v(t)dt$ et $\omega(t)dt$.

À partir du modèle choisi, les paramètres du mouvement et les profondeurs sont estimés par minimisation de l'erreur obtenue à partir de la contrainte d'illumination constante, à laquelle sont combinées d'éventuelles hypothèses sur les profondeurs

$$\sum_{x,y} \left(f(x,y) - g((x,y) + u(x,y,t)\ dt)\right)^2 .$$

Le champ de profondeurs est ainsi souvent supposé localement constant, par exemple par Horn et Weldon dans [33], et par Bergen dans [4]. Negahdaripour et Horn, dans [50], proposent une solution explicite au problème en supposant la surface plane ou quadratique. Dans [23], Ha et Kweon ajoutent un terme de régularisation préservant les discontinuités.

1.5.2 Méthodes discrètes

Ces méthodes sont très nombreuses ; on développera ici brièvement l'approche de matrice essentielle discrète [13, 15, 16, 34] et les techniques incrémentales de filtrage de Kalman [1, 77].

1.5.2.1 Estimation de la matrice essentielle

La contrainte épipolaire permet de lier les projections d'un point de l'espace sur deux images, indépendamment de la profondeur de ce point. Pour estimer le mouvement de la caméra entre ces deux images à partir de couples de points appariés, on peut estimer la matrice essentielle car elle suffit à retrouver la rotation et la translation [13]. Il est alors nécessaire de connaître les paramètres intrinsèques de la caméra. Dans le cas d'une caméra non calibrée, il existe des méthodes de détermination de la matrice fondamentale ; plusieurs d'entre elles sont comparées par Luong et coll. dans [44].

Comme la matrice essentielle E dépend exclusivement de la rotation et de la translation de la caméra, cinq paramètres suffisent à la déterminer complètement. L'algorithme des cinq points, proposé par Faugeras dans [13] nécessite cinq couples de points appariés menant à cinq équations de la forme de l'équation de Longuet-Higgins (1.15). L'algorithme des cinq points étant non-linéaire, une méthode linéaire moins complexe avait été proposée par Longuet-Higgins [42]. C'est l'algorithme des huit points, qui nécessite huit paires de points appariés. La translation et la rotation sont ensuite estimés en factorisant la matrice essentielle obtenue. Cependant, cet algorithme est très sensible au bruit ; beaucoup d'autres techniques plus robustes ont été développées [38, 13]. En renormalisant les coordonnées des points utilisés, Hartley [25] améliore très nettement les performances de l'algorithme des huit points.

Tsai et Huang ont montré dans [71] que seuls deux déplacements dans l'espace correspondent à une matrice essentielle donnée. Le nombre de couples de points appariés et le nombre de mouvements de caméra compatibles avec ces appariements est discuté dans [16]. Ces méthodes présentent l'avantage de ne supposer aucun *a priori* sur le type de mouvement de caméra et d'utiliser des techniques d'algèbre linéaire simples et rapides.

1.5.2.2 Méthodes incrémentales

Les méthodes précédentes sont bien adaptées à l'estimation d'un mouvement de caméra suffisamment important, entre des vues bien séparées d'une scène fixe, mais elles le sont moins pour traiter des images consécutives dans une séquence.

Une approche radicalement différente consiste à utiliser des techniques d'estimation récursives, considérant des mouvements incrémentés entre les images d'un film. Ces méthodes nécessitent la détection préalable et le suivi de points dans la séquence. Différentes approches de tracking sont décrites par Shi et Tomasi dans [63].

À partir des points suivis dans la séquence, le filtre de Kalman est utilisé, notamment par Azarbayejani et Pentland dans [1] puis Yao et Calway dans [77], pour estimer le mouvement de la caméra et les profondeurs des points suivis. Le filtre de Kalman est un estimateur récursif qui donne une estimation optimale, au sens des moindres carrés, d'un état dynamique à partir d'une séquence d'observations associées à l'état et en supposant des statistiques de bruit gaussiennes. On appelle état au temps t, noté $s(t)$, le mouvement de la caméra entre l'image 0 et l'image t et les profondeurs des points suivis dans le repère de la caméra au temps t. La dynamique des

états entre deux images consécutives dans la séquence est simple : la position de la caméra à un temps donné est supposée être la même qu'au temps précédent, à un bruit gaussien près, et les profondeurs des points demeurent les mêmes. L'équation suivante décrit l'évolution de l'état s

$$s(t) = Fs(t-1) + w(t)$$

où w est une bruit gaussien centré et F est la matrice de transition des états. En notant $z(t)$ les observations, c'est-à-dire les positions des points suivis sur l'image au temps t, un modèle d'observations est donné

$$z(t) = Hs(t) + v(t)$$

où v est un bruit gaussien centré et H la matrice d'observation.

Le principe général du filtre de Kalman est le suivant ; d'abord, une prédiction $\hat{s}(t)$ de $s(t)$ connaissant $\hat{s}(t-1)$ est effectuée en utilisant l'opérateur F. Puis, la prédiction est comparée aux observations des points au temps t, $z(t)$, ce qui fournit une erreur $e = z(t) - H\hat{s}(t)$, appelée innovation. Le gain de Kalman est alors calculé, proportionnel à la covariance de l'état prédit et inversement proportionnel à la covariance des observations. La prédiction de l'état au temps $\hat{s}(t)$ est mise à jour en ajoutant le produit de l'innovation et du gain. Plus le gain est grand, plus le filtre tient compte des observations, plus il est petit, et plus le filtre tient compte du modèle.

1.5.3 Méthodes différentielles

Dans le cas d'une caméra filmant une scène fixe, le flot optique de l'image dépend non linéairement de la distance de la caméra à chaque point de la scène (c'est-à-dire la profondeur des points dans le repère de la caméra), et des vitesses de rotation et translation de la caméra, ainsi que le montre la formule (1.9). Cette non-linéarité rend difficile le problème d'estimation du mouvement.

Parmi les méthodes utilisant le flot optique, trois approches sont présentées ici : la première est basée sur la contrainte linéaire (1.9) liant le flot optique aux vitesses angulaire et translationnelle, la deuxième sur la contrainte épipolaire différentielle et la troisième sur le mouvement de parallaxe.

Si le flot optique mesuré était exact, seuls quelques vecteurs suffiraient au calcul du mouvement de la caméra. Cependant, le flot mesuré est souvent très bruité et imprécis. Dans les méthodes décrites ci-après, des approches robustes des moindres carrés sont fréquemment mises en oeuvre pour exploiter toute l'information disponible.

Avant d'expliciter différentes techniques, notons que l'application qui, à un mouvement de caméra, associe un flot optique n'est pas injective. En effet, la translation de vecteur t d'un plan de l'espace fournit le même flot optique que la translation de vecteur αt du plan précédent dilaté d'un facteur α. La translation sera donc toujours estimée à un facteur d'échelle près. En revanche, deux rotations de caméra différentes ne généreront jamais le même flot optique [9]. L'injectivité du flot optique sera discutée plus précisément dans le chapitre 5.

Remarquons enfin que ces méthodes ne s'appliquent pas à des mouvements de caméra trop importants entre deux images (par exemple lorsque l'on ne considère plus deux images consécutives dans une séquence, mais éloignées dans le temps), car elles sont basées sur des approximations de mouvements instantanés.

1.5.3.1 Méthodes basées sur la contrainte (1.9)

L'objectif de ces méthodes est l'estimation des vitesses translationnelle $v(t)\,dt$ et angulaire $\omega(t)\,dt$, où dt est le pas de temps entre les acquisitions de deux images consécutives f et g, à partir de la donnée de n vecteurs de flot optique $\{u(x_i, y_i, t)\,dt\}_{i=1...n}$ mesurés aux points $\{(x_i, y_i)\}_{i=1...n}$ de f, en utilisant la contrainte (1.9) linéaire en $(\omega(t), v(t))$. En général, le nombre de vecteurs de flot connus est important, voire égal au nombre de points de l'image.

Pour alléger les notations, on note v, ω et $u(x_i, y_i)$ pour $v(t)\,dt$, $\omega(t)\,dt$ et $u(x_i, y_i, t)\,dt$. On appelle résidu $r(x_i, y_i)$ relatif au point (x_i, y_i) et au flot associé $u(x_i, y_i)$, le vecteur de $\mathbb{R}^2$

$$r(x_i, y_i) = u(x_i, y_i) - \frac{1}{Z(x_i, y_i)}\, A(x_i, y_i)\, v - B(x_i, y_i)\, \omega.$$

Le but de ces méthodes est la minimisation de la somme des normes des n résidus sur les vitesses v et ω. La difficulté de cette minimisation tient à l'ignorance des valeurs des profondeurs $\{Z(x_i, y_i)\}_{i=1...n}$.

Dans [9], Bruss et Horn, pionniers dans le sujet, éliminent la profondeur des résidus en observant que l'estimateur des moindres carrés de $Z(x_i, y_i)$ s'exprime en fonction des vitesses de la caméra ; l'équation

$$\frac{\partial \|r(x_i, y_i)\|^2}{\partial Z(x_i, y_i)} = 0$$

fournit l'expression de $Z(x_i, y_i)$. Ils ajoutent au problème de minimisation la contrainte $\|v\|^2 = 1$ en introduisant un multiplicateur de Lagrange. En différenciant en les sept paramètres, ils obtiennent un système de sept équations à sept inconnues. Parmi ces équations, trois sont linéaires par rapport aux trois composantes de ω. Ces composantes sont donc déterminées de façon unique et explicite en fonction des composantes de v. À partir des quatre autres équations polynomiales, cubiques ou quadratiques, les composantes de la vitesse v sont estimées par une méthode numérique.

Dans [80], Zucchelli, Santos-Victor et Christensen estiment simultanément les vitesses rotationnelle et translationnelle et la structure de la scène. La somme des normes des résidus dépend des paramètres inconnus rangés dans le vecteur p de taille $n+6$, $p = (v, \omega, Z(x_1, y_1), ..., Z(x_n, y_n))$. Les paramètres du mouvement et de la structure de la scène sont ici estimés par résolution du problème des moindres carrés

$$\hat{p} = \underset{p}{\operatorname{argmin}} \sum_{i=1}^{n} ||r(x_i, y_i)||^2. \tag{1.16}$$

Les auteurs utilisent la méthode itérative de Gauss-Newton. Pour améliorer cette estimation, ils proposent d'utiliser les contraintes géométriques de la scène. Pour cela, ils incorporent à l'équation (1.16) des informations de coplanarité et de colinéarité et utilisent la méthode de Levenberg-Marquardt pour résoudre le nouveau système. Ici, les vitesses de rotation et de translation sont évaluées simultanément avec les profondeurs des n points de l'image choisis.

Heeger et Jepson utilisent aussi la contrainte linéaire (1.9) ; ils estiment le mouvement et la structure de la scène en séparant, par des méthodes de sous-espaces linéaires, le problème en trois étapes ; dans [27], ils évaluent d'abord la translation, indépendamment de la rotation et des profondeurs des points choisis, puis la rotation et enfin les profondeurs. La structure de la scène est donc déterminée après l'estimation du mouvement. Pour cela, ils écrivent

$$\begin{aligned}\sum_{i=1}^{n} \|r(x_i,y_i)\|^2 &= \sum_{i=1}^{n} \left\| u(x_i,y_i) - \begin{pmatrix} A(x_i,y_i)v \\ B(x_i,y_i) \end{pmatrix}^T \begin{pmatrix} \dfrac{1}{Z(x_i,y_i)} \\ \omega \end{pmatrix} \right\|^2 \\ &= \sum_{i=1}^{n} \left\| u(x_i,y_i) - C(v,x_i,y_i)\, q(Z(x_i,y_i),\omega) \right\|^2\end{aligned}$$

où $C(v,x_i,y_i)$ est une matrice de taille 2×4 et $q(Z(x_i,y_i),\omega)$ est un vecteur de taille 4. Les auteurs montrent alors que la minimisation de $\sum_{i=1}^{n} ||r(x_i,y_i)||^2$ est équivalente à la minimisation de

$$E(v) = \sum_{i=1}^{n} \left\| C^{\perp}(v,x_i,y_i)\, u(x_i,y_i) \right\|^2$$

où $C^{\perp}(v,x_i,y_i)$ est le complément orthogonal de $C(v,x_i,y_i)$ c'est-à-dire $C^{\perp}(v,x_i,y_i)C(v,x_i,y_i) = 0$. L'expression à minimiser ne dépend plus que de la vitesse de déplacement v et on ne peut estimer que la direction de cette vitesse. L'espace des directions candidates est la demi-sphère unité. L'image de flot donnée est divisée en imagettes et chaque imagette fournit une surface résiduelle qui est l'image de $E(v)$ sur la demi-sphère unité. La somme des surfaces donne une estimation globale des moindres carrés de la vitesse v.

Une fois la translation estimée, la profondeur est exclue de l'équation (1.9) en multipliant cette équation par $d(x_i,y_i,v)$, vecteur unitaire orthogonal à la composante translationnelle du flot optique, c'est-à-dire

$$d^T(x_i,y_i,v)A(x_i,y_i)\,v = 0.$$

Ainsi, la contrainte (1.9) devient

$$d^T(x_i,y_i,v)\,u(x_i,y_i) = d^T(x_i,y_i,v)\,B(x_i,y_i)\,\omega.$$

La solution des moindres carrés pour ω est obtenue en minimisant

$$E(\omega) = \sum_{i=1}^{n} \left\| d^T(x_i,y_i,v)B(x_i,y_i)\ \omega - d^T(x_i,y_i,v)u(x_i,y_i) \right\|^2.$$

La profondeur des points $Z(x_i,y_i)$ peut alors être évaluée.

1.5.3.2 Méthodes basées sur la contrainte épipolaire différentielle

La contrainte épipolaire différentielle, équivalent continu de la contrainte épipolaire définie dans le paragraphe 1.4.4, lie les coordonnées d'un point de l'image et le flot optique mesuré en ce point aux vitesses de rotation et de translation de la caméra, sans que la profondeur du point projeté n'intervienne.

Soit M un point de l'espace. Supposons le mouvement de la caméra lisse dans le temps, de vitesses de translation $v(t)$ et de rotation $\omega(t)$. Cela revient à considérer que le point M se déplace dans l'espace à une vitesse $\dot{M}(t)$. Pour alléger les notations, on omet dans la suite la variable temporelle. La vitesse au point M est égale, d'après l'équation (1.10), à

$$\dot{M} = -[\omega]_\times M - v.$$

Soit m la projection de M sur le plan rétinien $\mathcal{R}$ et $\dot{m}$ le flot optique au point m. En multipliant l'équation précédente par $[v]_\times \mathbf{m}$, on obtient

$$\dot{M}^T [v]_\times \mathbf{m} = -M^T [\omega]_\times^T [v]_\times \mathbf{m} \qquad \text{car} \qquad v^T [v]_\times = 0.$$

Or, les coordonnées euclidiennes du point M sont proportionnelles aux coordonnées projectives du point m ; on peut donc remplacer M par $\lambda\mathbf{m}$ (λ non nul) et $\dot{M}$ par $\dot{\lambda}\mathbf{m} + \lambda\dot{\mathbf{m}}$

$$(\dot{\lambda}\mathbf{m} + \lambda\dot{\mathbf{m}})^T [v]_\times \mathbf{m} = -\lambda \mathbf{m}^T [\omega]_\times^T [v]_\times \mathbf{m}.$$

Comme $[v]_\times$ est une matrice antisymétrique, $\mathbf{m}^T [v]_\times \mathbf{m} = 0$ et on peut simplifier l'expression. Ainsi, si $\dot{m}$ est le flot optique au point m de l'image, généré par une caméra ayant une vitesse translationnelle v et angulaire ω, alors $\dot{m}$ et m vérifient

$$\begin{aligned} &\dot{\mathbf{m}}^T [v]_\times \mathbf{m} + \mathbf{m}^T [\omega]_\times^T [v]_\times \mathbf{m} = 0 \\ \Leftrightarrow\ &\dot{\mathbf{m}}^T [v]_\times \mathbf{m} - \mathbf{m}^T [\omega]_\times [v]_\times \mathbf{m} = 0. \end{aligned} \tag{1.17}$$

On appelle cette dernière équation contrainte épipolaire différentielle, en raison de sa ressemblance avec l'équation (1.14).

Dans [45], Ma, Kosecka et Sastry définissent le concept de matrice essentielle différentielle à partir de la contrainte épipolaire différentielle (1.17). En écrivant

$$[\omega]_\times [v]_\times = \frac{1}{2}\left([\omega]_\times [v]_\times - [v]_\times [\omega]_\times\right) + \frac{1}{2}\left([\omega]_\times [v]_\times + [v]_\times [\omega]_\times\right),$$

on obtient

$$\mathbf{m}^T [\omega]_\times [v]_\times \mathbf{m} = \mathbf{m}^T \frac{1}{2}\left([\omega]_\times [v]_\times + [v]_\times [\omega]_\times\right) \mathbf{m}$$

car la matrice $\frac{1}{2}\left([\omega]_\times [v]_\times - [v]_\times [\omega]_\times\right)$ est antisymétrique. L'équation (1.17) peut donc s'écrire matriciellement

$$\begin{pmatrix} \dot{\mathbf{m}}^T & \mathbf{m}^T \end{pmatrix} \begin{pmatrix} [v]_\times \\ -s \end{pmatrix} \mathbf{m} = 0 \tag{1.18}$$

où s est une matrice symétrique carrée d'ordre 3 définie par

$$s = \frac{1}{2}([\omega]_\times[v]_\times + [v]_\times[\omega]_\times).$$

La matrice $E = \begin{pmatrix} [v]_\times \\ -s \end{pmatrix}$ appartenant à $\mathcal{M}_{6,3}(\mathbb{R})$, est appelée matrice essentielle différentielle. L'ensemble de ces matrices forme l'espace essentiel différentiel $\mathcal{E}$ et l'ensemble des matrices s l'espace des matrices symétriques spéciales $\mathcal{S}$. La vitesse v et la matrice s sont estimées par minimisation des moindres carrés de l'équation (1.18). Les auteurs montrent que les vitesses v et ω sont ensuite obtenues de façon unique à partir de l'estimation de s et de l'estimateur des moindres carrés de v. Cette approche est généralisée au cas non calibré dans [8].

Dans [38, 39], Kanatani reformule la contrainte épipolaire différentielle avec le flot twisté $\dot{m}^*$, c'est-à-dire le flot $\dot{m}$ tourné de 90° autour de m

$$\dot{\mathbf{m}}^{*T}[v]_\times\mathbf{m} - \mathbf{m}^T\mathcal{K}\mathbf{m} = 0$$

où

$$\mathcal{K} = \omega^T v I_3 - \frac{1}{2}\left([\omega]_\times[v]_\times[v]_\times[\omega]_\times\right).$$

La vitesse v et la matrice $\mathcal{K}$ sont estimées en résolvant le problème linéaire des moindres carrés correspondants. La vitesse angulaire ω est extraite de l'estimation de la matrice $\mathcal{K}$ par

$$\omega = \frac{1}{2}\left(tr(\mathcal{K}) + 3v^T\mathcal{K}v\right)v - 2\mathcal{K}v.$$

1.5.3.3 Méthodes basées sur le mouvement de parallaxe

L'effet de parallaxe, décrit plus haut, a un effet gênant pour la mise en correspondance de points entre images mais c'est aussi un moyen d'estimer le mouvement de la caméra à partir du flot optique.

Supposons, dans un premier temps, que le mouvement de la caméra soit une translation pure, de vitesse $v(t) = (v_1(t), v_2(t), v_3(t))$. On connaît, dans ce cas, l'expression du flot au point (x, y) correspondant à la projection d'un point de profondeur Z avant le déplacement de la caméra, d'après la formule (1.9)

$$u(x, y, t) = \frac{1}{Z}\begin{pmatrix} -v_1(t) + x\ v_3(t) \\ -v_2(t) + y\ v_3(t) \end{pmatrix}.$$

Si on note P_0 le point de coordonnées $(x_0, y_0) = \left(\frac{v_1(t)}{v_3(t)}, \frac{v_2(t)}{v_3(t)}\right)$, alors

$$u(x, y, t) = \frac{v_3(t)}{Z}\begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}.$$

Ainsi, si $v_3(t)$ est non nulle, les vecteurs de flot optique sont dirigés vers le point P_0 ou au contraire envers lui. Autrement dit, le champ de flot est centré au point P_0, appelé foyer d'expansion (FOE

dans la littérature pour Focus Of Expansion). Dans le cas où $v_3(t)$ est nulle, tous les vecteurs ont même direction $(-v_1(t), -v_2(t))$.

Dans un deuxième temps, prenons un mouvement de caméra complet, composé d'une rotation suivie d'une translation. Dans [28], Helmholtz observe que la différence des flots optiques en deux points très proches dans l'image mais correspondant à des profondeurs différentes dans l'espace, est quasiment indépendante de la rotation et ne dépend donc pratiquement que de la différence des inverses des deux profondeurs. Ainsi, une discontinuité de profondeur dans l'espace, du point de vue de la caméra, correspond sur l'image à une discontinuité dans la composante translationnelle du flot optique. À partir d'un flot optique donné, si on construit un champ de vecteurs représentant les différences des vecteurs dans un petit voisinage, les vecteurs différence seront orientés vers ou envers le foyer d'expansion le long des discontinuités de profondeur.

Différentes méthodes d'estimation du mouvement de la caméra sont basées sur le mouvement de parallaxe. Longuet-Higgins, dans [42], utilise les dérivées spatiales du flot optique pour identifier le foyer d'expansion et donc la direction de la translation. Cependant, la méthode est très sensible au bruit présent dans le champ de vecteurs. Rieger et Lawton, dans [57], calculent les différences d'un vecteur de flot avec les flots d'un voisinage. Puis, à partir de la distribution des flots, l'orientation dominante des vecteurs de différence est calculée. Seules les orientations des vecteurs résultants situés aux points où la distribution des vecteurs est très anisotrope, sont conservées. De tels points apparaissent en effet au niveau de variations de profondeurs importantes. Les auteurs obtiennent ainsi un champ correspondant au champ de vitesse translationnelle, ce qui permet de déduire le foyer d'expansion. Lorsque la direction de la translation est connue, les composantes du flot orthogonales à cette direction ne peuvent qu'être dues à la rotation. Ce modèle a été amélioré par Hildreth [29].

Une utilisation différente du mouvement de parallaxe est présentée par Tomasi et Shi dans [70]. La translation est estimée à partir des déformations d'images. À partir d'un couple (m, m') de points dans une image, suivis dans l'image suivante, la déformation est mesurée par la variation dans le temps $\dot{a}$ de l'angle $a = \arccos(m, m')$. La variation de l'angle a étant indépendante de la rotation, elle dépend des profondeurs Z et Z' des points projetés en m et m' et de la translation t

$$\dot{a} = \sin a \begin{pmatrix} \frac{1}{Z'} \\ \\ \frac{1}{Z} \\ \\ 0 \end{pmatrix}^T \begin{pmatrix} m \\ m' \\ \frac{m \wedge m'}{\|m \wedge m'\|} \end{pmatrix}^{-1} t.$$

Pour un sous-ensemble de n points suivis entre deux images (grâce au flot optique par exemple), on aboutit à une combinaison des contraintes ci-dessus à minimiser sur les composantes de la translation et les n profondeurs des points. Dans [70], la minimisation est réalisée par une méthode de projection des variables sur la sphère unité $t = 1$.

1.5.4 Conclusion sur les méthodes présentées

Les méthodes d'estimation d'un mouvement de caméra donnent rarement de bons résultats à la fois dans les situations de mouvements importants et de mouvements faibles. Les méthodes discrètes sont adaptées à l'estimation de grands mouvements, c'est-à-dire à des images éloignées dans la séquence. Les méthodes différentielles, au contraire, basées sur des approximations infinitésimales, donnent de bonnes estimations de petits mouvements. Les méthodes directes sont elles aussi adaptées à de faibles mouvements de caméra car elles reposent sur l'équation de contrainte du flot optique. Cependant, il est aussi possible d'estimer des mouvements de caméra plus conséquents avec ces méthodes (jusqu'à 10 à 15% de la taille de l'image) en utilisant un modèle de mouvement adapté et un traitement multirésolution.

Pour évaluer les performances des méthodes d'estimation, on considère principalement deux critères. La robustesse d'abord ; les algorithmes ne doivent pas être trop sensibles aux erreurs sur les mesures de flot optique ou sur la précision des appariements. En général, les algorithmes locaux et les méthodes basées sur des approximations sont plus sensibles que les algorithmes globaux et ceux basés sur des formules exactes. Plus les méthodes utilisent d'appariements, plus elles sont robustes. Le deuxième critère est l'efficacité ; les algorithmes qui mettent en oeuvre des techniques numériques itératives dans un espace de solution à grande dimension coûtent cher en temps de calcul. Pour les méthodes différentielles, la détermination préalable du flot optique sur une séquence est souvent coûteuse, ce qui pénalise le coût global des méthodes.

Une comparaison de six algorithmes utilisant le flot optique pour l'estimation du mouvement de caméra est présentée par Tian, Tomasi et Heeger dans [69]. Des critères quantifiant la sensibilité au bruit des méthodes et leur convergence (lorsqu'elles mettent en oeuvre une recherche numérique) sont établis. Pour cela, des nuages de points sont aléatoirement générés dans l'espace tridimensionnel devant la caméra. Divers mouvements de caméra sont choisis et le flot optique résultant est calculé par la formule (1.9). Un bruit gaussien en quantité variable est ajouté à chaque composante du flot. Les auteurs utilisent ces données pour tester et comparer les algorithmes : la méthode de Bruss et Horn [9] se révèle être la plus robuste au bruit.

Chapitre 2

Déformations produites par un mouvement de caméra

L'objet de ce chapitre est la description des déformations observées entre deux images consécutives dans une séquence. Nous précisons tout d'abord le contexte permettant d'approximer la profondeur de la scène par une constante dans les expressions des déformations. Dans ce cadre, nous choisissons de modéliser les déformations dans un groupe, le groupe des recalages, introduit par F. Dibos dans [11], isomorphe au groupe $SE(3)$ des déplacements de l'espace. À la différence du groupe projectif, ce groupe permet de composer et d'inverser les déformations en les associant à des mouvements de caméra. Nous présentons alors une nouvelle décomposition d'un mouvement dans l'espace, permettant de séparer la déformation entre deux images consécutives en deux composantes : une similitude et une déformation "purement" projective. Cette nouvelle écriture du mouvement conduit aussi à une approximation quadratique des déformations, somme de termes indépendamment associés aux différents éléments de la décomposition proposée.

2.1 Contexte

Dans ce chapitre, on considère une caméra de longueur focale unitaire, filmant une scène fixe, et deux images consécutives dans la séquence vidéo obtenue. On écrit le mouvement de caméra entre les deux images $D = (R, t)$ dans le groupe Spécial Euclidien $SE(3)$ défini dans le chapitre 1 et on note la matrice de rotation orthonormale

$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$$

et le vecteur de translation

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}.$$

Les deux images consécutives obtenues avant et après le déplacement de la caméra sont notées f et g. Elles sont respectivement définies sur un domaine K du plan rétinien $\{Z = 1\}$ et un domaine K' du plan $\{Z' = 1\}$ (plan rétinien associé à la caméra après son déplacement). Ces deux domaines sont rectangulaires et ont mêmes dimensions.

Le nombre d'images par seconde acquises par la caméra étant élevé, classiquement 24, le déplacement de la caméra entre deux acquisitions consécutives est très réduit, même si la vitesse est importante. En conséquence, les deux images obtenues sont très proches.

Rappelons les relations exactes entre f et g. Soit (x, y) un point de K et (x', y') un point de K', tous deux projections d'un même point de l'espace. On considère dans ce document l'éclairement constant ; on a donc $f(x, y) = g(x', y')$. Soit $Z(x, y)$ la profondeur du point de l'espace projeté en (x, y) sur K, donnée dans le repère de la caméra avant le déplacement, et $Z'(x', y')$ la profondeur de ce même point de l'espace, exprimée dans le repère de la caméra après le déplacement. Les fonctions Z et Z' définies sur K et K' sont supposées strictement positives. La formule (1.11), reprise ici, fournit l'expression de (x', y') en fonction de (x, y) et de la profondeur $Z(x, y)$

$$\begin{cases} x' = \dfrac{a_1 x + a_2 y + a_3 - \langle \frac{t}{Z(x,y)}, R(i) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z(x,y)}, R(k) \rangle} \\ \\ y' = \dfrac{b_1 x + b_2 y + b_3 - \langle \frac{t}{Z(x,y)}, R(j) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z(x,y)}, R(k) \rangle}, \end{cases} \tag{2.1}$$

et la formule (1.12) l'expression de (x, y) en fonction de (x', y') et de la profondeur $Z'(x', y')$

$$\begin{cases} x = \dfrac{a_1 x' + b_1 y' + c_1 + \frac{t_1}{Z'(x',y')}}{a_3 x' + b_3 y' + c_3 + \frac{t_3}{Z'(x',y')}} \\ \\ y = \dfrac{a_2 x' + b_2 y' + c_2 + \frac{t_2}{Z'(x',y')}}{a_3 x' + b_3 y' + c_3 + \frac{t_3}{Z'(x',y')}}. \end{cases} \tag{2.2}$$

2.1.1 Hypothèses sur le flot optique

2.1.1.1 Taille des images

Soit L la plus grande dimension des domaines K et K'. La variable L est finie car l'angle de vue de la caméra est limité. En pratique, il dépasse rarement 150° (une caméra dont l'angle de vue vaut 150° est appelée caméra grand-angle).

Pour une longueur focale égale à 1, l'angle de vue a d'une caméra et la taille des images sont liés par la relation

$$\tan\left(\frac{a}{2}\right) = \frac{L}{2},$$

comme illustré sur la figure (2.1).

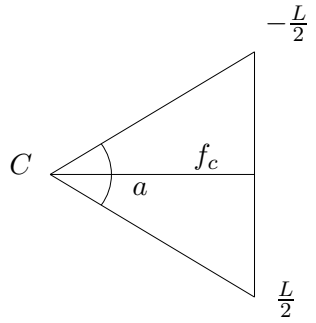

FIGURE 2.1: *Rapport entre l'angle de vue a d'une caméra, la dimension L des images et la longueur focale f_c.*

Par exemple, la condition $a \leq 150°$ implique

$$L \leq 8.$$

2.1.1.2 Flot optique

Les différences entre deux images consécutives dans une séquence sont très faibles ; nous allons ici les quantifier. Nous avons obtenu les résultats donnés ci-après expérimentalement. On considère

- une caméra de longueur focale égale à 1
- un déplacement $D = (R, t) \in SE(3)$ entre deux images consécutives.

Définition 2.1 – *On appelle grossissement associé au mouvement $D = (R, t)$ et à la fonction de profondeur Z définie sur K*

$$G_{D,Z} = \max_{(x,y)\in K} \Big| \frac{1}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z(x,y)}, R(k) \rangle} \Big|.$$

Hypothèse 1 – *Il existe G_{max} tel que pour tout déplacement $D = (R, t)$ entre deux images consécutives et toute fonction de profondeur Z associée à la scène*

$$G_{D,Z} \leq G_{max}.$$

Hypothèse 2 – *Soient $D = (R, t) \in SE(3)$ et K le domaine rectangulaire de plus grande dimension L sur lequel l'image f est définie. Soit Z la fonction définie sur K donnant les profondeurs des points de $\mathbb{R}^3$ projetés sur K par une caméra de longueur focale unitaire. Pour un point $(x, y) \in K$, on note (x', y') le point apparié sur K' par la formule (2.1). On suppose*

$$\max_{(x,y)\in K} \big\{ |x' - x|, |y' - y| \big\} \leq \frac{L}{2}.$$

L'hypothèse (1) provient du fait que le déplacement de l'axe optique de k à $R(k)$ est nécessairement très faible pour que les images soient exploitables. En effet,

$$c_1x + c_2y + c_3 - \langle \frac{t}{Z(x,y)}, R(k)\rangle = c_1\left(x - \frac{t_1}{Z(x,y)}\right) + c_2\left(y - \frac{t_2}{Z(x,y)}\right) + c_3\left(1 - \frac{t_3}{Z(x,y)}\right)$$

où $(c_1, c_2, c_3) = R(k)$. On peut paramétrer $R(k)$ par deux angles θ et α, comme décrit sur la figure (2.2), et on a

$$R(k) = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} \sin\theta\sin\alpha \\ -\cos\theta\sin\alpha \\ \cos\alpha \end{pmatrix}.$$

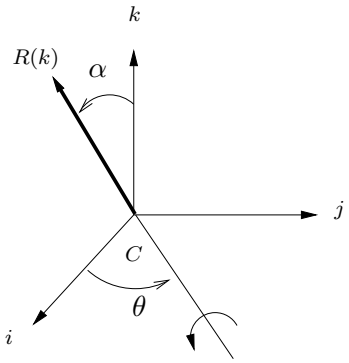

FIGURE 2.2: *Paramétrage de $R(k)$ par θ et α.*

On obtient alors

$$c_1x + c_2y + c_3 - \langle \tfrac{t}{Z(x,y)}, R(k)\rangle =$$

$$\cos\alpha\left(1 - \tfrac{t_3}{Z(x,y)}\right) + \sin\alpha\left(\sin\theta\left(x - \tfrac{t_1}{Z(x,y)}\right) + \cos\theta\left(y - \tfrac{t_2}{Z(x,y)}\right)\right).$$

L'angle α étant très faible, l'expression $1/\left(c_1x + c_2y + c_3 - \langle \frac{t}{Z(x,y)}, R(k)\rangle\right)$ est très proche de $1/\left(1 - \frac{t_3}{Z(x,y)}\right)$, correspondant au rapport de l'homothétie générée sur les images par la translation de la caméra le long de son axe optique. En toute généralité, la constante G_{max} est donc supérieure à 1. En pratique, $G_{max} = 4/3$ est une valeur raisonnable pour un mouvement de caméra entre deux acquisitions consécutives d'image.

L'hypothèse (2) exprime la restriction de l'amplitude des déplacements apparents des points entre deux images consécutives. On suppose qu'un point ne peut se déplacer d'une longueur supérieure à la moitié de la taille de l'image selon les directions horizontale et verticale. Cette hypothèse est tout à fait réaliste car la majoration par $L/2$ est très large.

2.1.2 Approximation des profondeurs par une profondeur uniforme

On souhaiterait approximer les profondeurs par une constante dans les formules (2.1) et (2.2). Soit $Z_0 \in \mathbb{R}_+^*$. Par un développement limité de Taylor à l'ordre 1 des formules (2.1) en $\frac{1}{Z(x,y)}$ au voisinage de $\frac{1}{Z_0}$, on obtient

$$\left\{\begin{array}{ll} x' = & \dfrac{a_1x+a_2y+a_3-\langle\frac{t}{Z_0},R(i)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle}+ \\ & \left(\frac{1}{Z(x,y)}-\frac{1}{Z_0}\right)\left(-\langle t,R(i)\rangle+\langle t,R(k)\rangle\frac{a_1x+a_2y+a_3}{\left(c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle\right)^2}\right)+\ o\left(\frac{1}{Z(x,y)}-\frac{1}{Z_0}\right) \\ y' = & \dfrac{b_1x+b_2y+b_3-\langle\frac{t}{Z_0},R(j)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle}+ \\ & \left(\frac{1}{Z(x,y)}-\frac{1}{Z_0}\right)\left(-\langle t,R(j)\rangle+\langle t,R(k)\rangle\frac{b_1x+b_2y+b_3}{\left(c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle\right)^2}\right)+\ o\left(\frac{1}{Z(x,y)}-\frac{1}{Z_0}\right). \end{array}\right.$$

Par conséquent, si pour tout $(x,y) \in K$, $\left(\frac{1}{Z(x,y)}-\frac{1}{Z_0}\right)\|t\|$ est suffisamment petit, on peut approximer les profondeurs par Z_0 dans les formules (2.1). Plus précisément, le théorème suivant propose une condition suffisante pour approximer les formules à ε près pour tout $(x,y) \in K$.

Théorème 2.1 – *Soient une caméra de longueur focale unitaire, $D = (R,t) \in SE(3)$ avec $t \neq 0$, et K le domaine rectangulaire de plus grande dimension L sur lequel l'image f est définie. Soit Z la fonction définie sur K donnant les profondeurs des points de $\mathbb{R}^3$ projetés sur K. On note $Z_{inf} > 0$ et Z_{sup} les bornes de Z (finies ou infinies). On suppose que Z et D vérifient les hypothèses (1) et (2). Si*

$$\left(\frac{1}{Z_{inf}}-\frac{1}{Z_{sup}}\right)\|t\|\,(L+1)\,G_{max} \leq 2\varepsilon$$

alors, quel que soit Z_0 strictement positif vérifiant

$$\frac{1}{Z_{inf}}-\frac{\varepsilon}{\|t\|\,(L+1)\,G_{max}} \leq \frac{1}{Z_0} \leq \frac{1}{Z_{sup}}+\frac{\varepsilon}{\|t\|\,(L+1)\,G_{max}}$$

on peut écrire, $\forall (x,y) \in K$,

$$\left\{\begin{array}{l} \left|\dfrac{a_1x+a_2y+a_3-\langle\frac{t}{Z(x,y)},R(i)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z(x,y)},R(k)\rangle}-\dfrac{a_1x+a_2y+a_3-\langle\frac{t}{Z_0},R(i)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle}\right| \leq \varepsilon \\ \left|\dfrac{b_1x+b_2y+b_3-\langle\frac{t}{Z(x,y)},R(j)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z(x,y)},R(k)\rangle}-\dfrac{b_1x+b_2y+b_3-\langle\frac{t}{Z_0},R(j)\rangle}{c_1x+c_2y+c_3-\langle\frac{t}{Z_0},R(k)\rangle}\right| \leq \varepsilon. \end{array}\right.$$

Démonstration. Soit $Z_0 > 0$ tel que $1/Z_0 \in [1/Z_{inf} - \varepsilon/(\|t\|(L+1)G_{max}), 1/Z_{sup} + \varepsilon/(\|t\|(L+1)G_{max})]$ et $(x,y) \in K$. On note $\delta = \frac{1}{Z(x,y)} - \frac{1}{Z_0}$. En partant de la formule (2.1), on a

$$\begin{cases} x' = \dfrac{a_1x + a_2y + a_3 - \langle \frac{t}{Z_0}, R(i)\rangle - \delta\langle t, R(i)\rangle}{c_1x + c_2y + c_3 - \langle \frac{t}{Z_0}, R(k)\rangle - \delta\langle t, R(k)\rangle} = \dfrac{u_0^1 - \delta\langle t, R(i)\rangle}{v_0 - \delta\langle t, R(k)\rangle} \\ \\ y' = \dfrac{b_1x + b_2y + b_3 - \langle \frac{t}{Z_0}, R(j)\rangle - \delta\langle t, R(j)\rangle}{c_1x + c_2y + c_3 - \langle \frac{t}{Z_0}, R(k)\rangle - \delta\langle t, R(k)\rangle} = \dfrac{u_0^2 - \delta\langle t, R(j)\rangle}{v_0 - \delta\langle t, R(k)\rangle}. \end{cases}$$

On cherche à borner $\left|x' - \frac{u_0^1}{v_0}\right|$ et $\left|y' - \frac{u_0^2}{v_0}\right|$. En appliquant la formule de Taylor avec reste intégral en δ au voisinage de 0, on obtient

$$\begin{cases} x' = \dfrac{u_0^1}{v_0} + \displaystyle\int_0^\delta \frac{\langle t, R(k)\rangle\, u_0^1 - \langle t, R(i)\rangle\, v_0}{(v_0 - z\,\langle t, R(k)\rangle)^2}\, dz \\ \\ y' = \dfrac{u_0^2}{v_0} + \displaystyle\int_0^\delta \frac{\langle t, R(k)\rangle\, u_0^2 - \langle t, R(j)\rangle\, v_0}{(v_0 - z\,\langle t, R(k)\rangle)^2}\, dz \end{cases}$$

soit

$$\begin{cases} x' = \dfrac{u_0^1}{v_0} + \delta\, \dfrac{\langle t, R(k)\rangle\, u_0^1 - \langle t, R(i)\rangle\, v_0}{v_0\,(v_0 - \delta\,\langle t, R(k)\rangle)} \\ \\ y' = \dfrac{u_0^2}{v_0} + \delta\, \dfrac{\langle t, R(k)\rangle\, u_0^2 - \langle t, R(j)\rangle\, v_0}{v_0\,(v_0 - \delta\,\langle t, R(k)\rangle)}. \end{cases}$$

Or,

$$\begin{cases} \left|\dfrac{\langle t, R(k)\rangle\, u_0^1 - \langle t, R(i)\rangle\, v_0}{v_0\,(v_0 - \delta\,\langle t, R(k)\rangle)}\right| \leq \|t\|\, \dfrac{|u_0^1| + |v_0|}{|v_0|}\, \left|\dfrac{1}{v_0 - \delta\,\langle t, R(k)\rangle}\right| \\ \\ \left|\dfrac{\langle t, R(k)\rangle\, u_0^2 - \langle t, R(j)\rangle\, v_0}{v_0\,(v_0 - \delta\,\langle t, R(k)\rangle)}\right| \leq \|t\|\, \dfrac{|u_0^2| + |v_0|}{|v_0|}\, \left|\dfrac{1}{v_0 - \delta\,\langle t, R(k)\rangle}\right|. \end{cases}$$

Comme $(x,y) \in K \subseteq \left[-\frac{L}{2}, \frac{L}{2}\right]^2$ et les distances $\left|x - \frac{u_0^1}{v_0}\right|$ et $\left|y - \frac{u_0^2}{v_0}\right|$ sont bornées par $L/2$ d'après l'hypothèse (2), on a

$$\begin{cases} \left|\dfrac{u_0^1}{v_0}\right| + 1 \leq \left|\dfrac{u_0^1}{v_0} - x\right| + |x| + 1 \leq L + 1 \\ \\ \left|\dfrac{u_0^2}{v_0}\right| + 1 \leq \left|\dfrac{u_0^2}{v_0} - y\right| + |y| + 1 \leq L + 1, \end{cases}$$

et

$$\begin{cases} \left(\left|\frac{u_0^1}{v_0}\right| + 1\right) \frac{1}{\left|v_0 - \delta \langle t, R(k)\rangle\right|} \leq (L+1)\, G_{max} \\ \left(\left|\frac{u_0^2}{v_0}\right| + 1\right) \frac{1}{\left|v_0 - \delta \langle t, R(k)\rangle\right|} \leq (L+1)\, G_{max}, \end{cases}$$

d'où

$$\begin{cases} \left|x' - \frac{u_0^1}{v_0}\right| \leq |\delta|\, \|t\|\, (L+1)\, G_{max} \\ \left|y' - \frac{u_0^2}{v_0}\right| \leq |\delta|\, \|t\|\, (L+1)\, G_{max}. \end{cases}$$

Comme

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\|\, (L+1)\, G_{max} \leq 2\varepsilon$$

et en utilisant l'hypothèse (1)

$$\frac{1}{Z_{inf}} - \frac{\varepsilon}{\|t\|(L+1)\, G_{max}} \leq \frac{1}{Z_0} \leq \frac{1}{Z_{sup}} + \frac{\varepsilon}{\|t\|(L+1)\, G_{max}},$$

on a

$$\forall (x,y) \in K, \quad \left|\frac{1}{Z(x,y)} - \frac{1}{Z_0}\right| \|t\|\, (L+1)\, G_{max} \leq \varepsilon,$$

ce qui implique

$$\forall (x,y) \in K, \quad \begin{cases} \left|x' - \frac{u_0^1}{v_0}\right| \leq \varepsilon \\ \left|y' - \frac{u_0^2}{v_0}\right| \leq \varepsilon. \end{cases}$$

□

Remarques

- Pour une taille d'image fixée, on peut approximer les profondeurs des points projetés sur K par une constante dans les formules (2.1) si le produit des variations de $1/Z$ par la norme de la translation est suffisamment faible. Cette approximation revient à considérer la scène filmée plane et orthogonale à l'axe optique de la caméra avant le déplacement.
- Si $t = 0$, alors la profondeur des points projetés n'intervient pas dans les formules d'appariements.

– Si $t \neq 0$ et L sont fixés et

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} \leq 2\varepsilon,$$

les variations autorisées de la profondeur Z sont d'autant plus grandes que la scène est éloignée de la caméra.

– Si $t \neq 0$ et $Z_{sup} = +\infty$, alors la condition d'approximation des formules (2.1) à ε près, donnée dans le théorème, devient

$$\frac{\|t\|}{Z_{inf}} (L+1) \, G_{max} \leq 2\varepsilon.$$

Si toutes les profondeurs sont infinies, la condition est évidemment vérifiée. Mais à $\|t\|$ fixée, dans le cas d'une scène constituée d'objets placés à des distances finies de la caméra sur un fond de profondeur infinie (le ciel par exemple), les objets doivent être suffisamment éloignés pour que la condition soit vérifiée.

– Le choix optimal de Z_0 est celui donné par

$$\widehat{Z}_0 = \arg\min_{Z_0} \max_{(x,y) \in K} \left| \frac{1}{Z(x,y)} - \frac{1}{Z_0} \right|$$

soit

$$\frac{1}{\widehat{Z}_0} = \frac{1}{2} \left(\frac{1}{Z_{inf}} + \frac{1}{Z_{sup}} \right).$$

Ce choix minimise la borne supérieure sur K de l'erreur d'approximation des formules (2.1) lorsque l'on substitue Z_0 à $Z(x,y)$.

De façon analogue, on peut approximer les formules (2.2) à ε près en substituant une constante à $Z'(x',y')$. On voudrait maintenant approximer les profondeurs Z et Z' dans les formules (2.1) et (2.2) par une même constante.

Proposition 2.1 – *Soient une caméra de longueur focale unitaire, $D = (R,t) \in SE(3)$ avec $t \neq 0$, et K et K' les domaines rectangulaires de plus grande dimension L sur lesquels les images f et g sont définies. Soient Z et Z' les fonctions définies sur K et K', profondeurs des points projetés respectivement sur K et K'. On note Z_{inf} la borne inférieure (strictement positive) de Z. On suppose que Z et D vérifient l'hypothèse (1). Si $Z_0 > 0$ vérifie*

$$\forall (x,y) \in K, \quad \left| \frac{1}{Z(x,y)} - \frac{1}{Z_0} \right| \|t\| \, (L+1) \, G_{max} \leq \varepsilon$$

alors

$$\forall (x',y') \in K', \quad \left| \frac{1}{Z'(x',y')} - \frac{1}{Z_0} \right| \leq \frac{\varepsilon}{\|t\|(L+1) \, G_{max}} + \frac{G_{max} - 1}{Z_{inf}}.$$

Démonstration. Soient (x,y) et (x',y') deux points de K et K' appariés. On a

$$\left| \frac{1}{Z'(x',y')} - \frac{1}{Z_0} \right| \leq \left| \frac{1}{Z'(x',y')} - \frac{1}{Z(x,y)} \right| + \left| \frac{1}{Z(x,y)} - \frac{1}{Z_0} \right|.$$

Or, la profondeur $Z'(x',y')$ est liée à $Z(x,y)$ d'après (1.13) par

$$\begin{aligned} Z'(x',y') &= Z(x,y)\,(c_1x + c_2y + c_3) - \langle t, R(k)\rangle \\ &= Z(x,y)\left(c_1(x - \tfrac{t_1}{Z(x,y)}) + c_2(y - \tfrac{t_2}{Z(x,y)}) + c_3(1 - \tfrac{t_3}{Z(x,y)})\right) \end{aligned}$$

d'où d'après l'hypothèse (1)

$$\frac{1}{Z'(x',y')} \leq \frac{1}{Z(x,y)}\,G_{max}$$

donc

$$\left|\frac{1}{Z'(x',y')} - \frac{1}{Z(x,y)}\right| \leq \frac{G_{max}-1}{Z(x,y)} \leq \frac{G_{max}-1}{Z_{inf}}.$$

Finalement,

$$\left|\frac{1}{Z'(x',y')} - \frac{1}{Z_0}\right| \leq \frac{\varepsilon}{\|t\|(L+1)\,G_{max}} + \frac{G_{max}-1}{Z_{inf}}.$$

□

En conséquence, si la scène filmée est suffisamment éloignée de la caméra, les profondeurs d'un même point de l'espace projeté avant et après le déplacement sont très proches, comme illustré sur la figure (2.3).

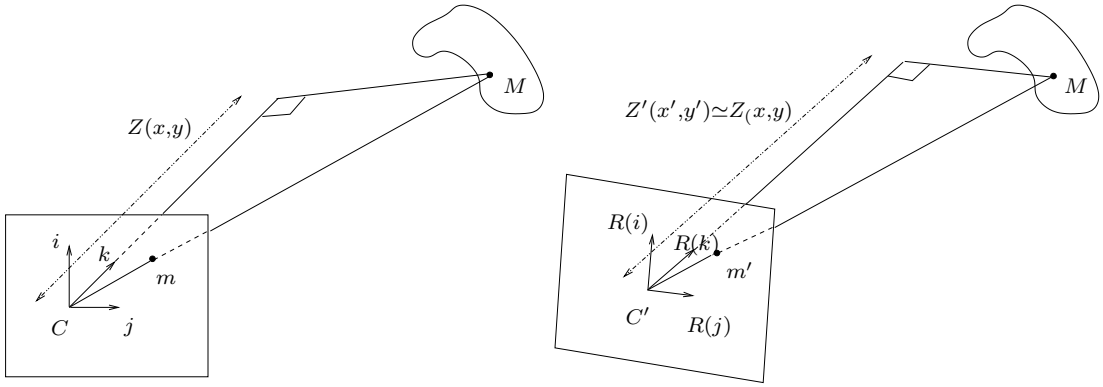

FIGURE 2.3: *Déplacement d'une caméra (entre deux acquisitions d'images successives) lorsque la scène filmée est éloignée du centre optique.*

En conclusion, dans le cas où $t \neq 0$, si

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right)\|t\|\,(L+1)\,G_{max} \leq 2\varepsilon,$$

on peut approximer les composantes de la formule (2.1) à ε près en remplaçant les profondeurs Z par une constante. La valeur ε peut être rendue d'autant plus faible que les variations de l'inverse de la profondeur ne sont pas trop importantes, la translation est suffisamment petite et la taille de l'image est limitée. Si on a aussi

$$\frac{\|t\|}{Z_{inf}}\,G_{max}\,(G_{max}-1)\,(L+1) \leq \varepsilon',$$

on peut à la fois approximer les formules (2.1) à ε près et les formules (2.2) à $\varepsilon + \varepsilon'$ près en remplaçant les profondeurs Z et Z' par une même constante. La valeur ε' peut être rendue d'autant plus petite que la caméra est éloignée de la scène (et que Z_{inf} est donc grand). Dans la suite du chapitre, nous allons supposer ε et ε' petits, ce qui nous permet de remplacer les profondeurs Z et Z' par une constante et de considérer la deuxième image comme une déformation de la première (et inversement). Ainsi, le problème d'estimation du mouvement de la caméra à partir d'images consécutives est simplifié ; on recherche des transformations planes afin de déterminer les paramètres du mouvement. En coordonnées projectives, ces applications sont linéaires, comme précisé dans le chapitre 1, et elles sont appelées homographies. Il existe des méthodes d'estimation d'homographies entre deux images à partir d'appariements de points ; Vincent et Laganière dans [73] proposent par exemple une méthode utilisant un schéma RANSAC (Random Sample Consensus) à partir de coins détectés dans les deux images. Cependant, une homographie dépend de huit paramètres et un mouvement de caméra seulement de six ; toute homographie ne peut donc être associée à un mouvement de caméra.

2.2 Groupe des recalages

2.2.1 Déformations projectives

Dans le contexte défini précédemment, nous considérons les profondeurs de la scène filmée avant et après le déplacement de la caméra égales à une constante Z_0. Deux points appariés (x, y) et (x', y') de K et K' sont donc liés par les relations

$$\begin{cases} x' = \dfrac{a_1 x + a_2 y + a_3 - \langle \frac{t}{Z_0}, R(i) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z_0}, R(k) \rangle} \\ \\ y' = \dfrac{b_1 x + b_2 y + b_3 - \langle \frac{t}{Z_0}, R(j) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z_0}, R(k) \rangle} \end{cases} \tag{2.3}$$

et

$$\begin{cases} x = \dfrac{a_1 x' + b_1 y' + c_1 + \frac{t_1}{Z_0}}{a_3 x' + b_3 y' + c_3 + \frac{t_3}{Z_0}} \\ \\ y = \dfrac{a_2 x' + b_2 y' + c_2 + \frac{t_2}{Z_0}}{a_3 x' + b_3 y' + c_3 + \frac{t_3}{Z_0}}. \end{cases} \tag{2.4}$$

Ainsi, les déformations observées entre les images f et g, et réciproquement entre les images g et f dépendent de la profondeur Z_0 de la scène. Mais cette profondeur n'affecte que les coefficients de la translation ; plus la scène filmée est éloignée de la caméra, moins les points projetés sont translatés. Les équations précédentes montrent qu'il est vain de chercher à estimer, à partir de f et g, à la fois la profondeur de la scène et la translation de la caméra. En effet, si le quadruplet (t_1, t_2, t_3, Z_0) est solution des équations (2.3) et (2.4), liant (x, y) à (x', y') et (x', y') à (x, y),

alors $(\lambda t_1, \lambda t_2, \lambda t_3, \lambda Z_0)$, λ étant positif (car seuls les points de profondeurs positives dans le repère de la caméra sont visibles par la caméra) l'est aussi. Nous estimerons donc seulement la direction de la translation ; nous noterons $\widetilde{t} = (\widetilde{t_1}, \widetilde{t_2}, \widetilde{t_3})$ la translation t divisée par la profondeur de la scène Z_0, correspondant à l'homothétie-translation observée entre l'image f et l'image g.

On a donc

$$g(x', y') = f\left(\frac{a_1x' + b_1y' + c_1 + \widetilde{t_1}}{a_3x' + b_3y' + c_3 + \widetilde{t_3}}, \frac{a_2x' + b_2y' + c_2 + \widetilde{t_2}}{a_3x' + b_3y' + c_3 + \widetilde{t_3}}\right)$$

que l'on écrit

$$g(x', y') = f \circ \varphi(x', y'). \tag{2.5}$$

et

$$f(x, y) = g\left(\frac{a_1x + a_2y + a_3 - \langle\widetilde{t}, R(i)\rangle}{c_1x + c_2y + c_3 - \langle\widetilde{t}, R(k)\rangle}, \frac{b_1x + b_2y + b_3 - \langle\widetilde{t}, R(j)\rangle}{c_1x + c_2y + c_3 - \langle\widetilde{t}, R(k)\rangle}\right)$$

notée

$$f(x, y) = g \circ \psi(x, y). \tag{2.6}$$

Définition 2.2 – *On appelle application projective ϕ de $\mathbb{R}^2$ dans $\mathbb{R}^2$ toute application associée à un automorphisme de $\mathbb{R}^3$, c'est-à-dire à une matrice $\mathcal{M}_\phi$ de $GL(3, \mathbb{R})$ par le diagramme commutatif*

$$\begin{array}{ccc} \mathbb{R}^3 & \xrightarrow{\mathcal{M}_\phi} & \mathbb{R}^3 \\ \pi \downarrow & & \downarrow \pi \\ \mathbb{R}^2 & \xrightarrow{\phi} & \mathbb{R}^2 \end{array}$$

où π est la projection sur le plan $\{Z = 1\}$ $\left\{\begin{array}{ccc} \pi : \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ (X,Y,Z) & \longmapsto & \left(\frac{X}{Z}, \frac{Y}{Z}\right) \end{array}\right.$.

Ainsi, à une matrice inversible $M = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix}$ est associée l'application projective ϕ

$$\begin{array}{ccc} \mathbb{R}^2 & \longrightarrow & \mathbb{R}^2 \\ (x, y) & \longmapsto & \phi(x, y) = \left(\frac{\alpha_1x + \beta_1y + \gamma_1}{\alpha_3x + \beta_3y + \gamma_3}, \frac{\alpha_2x + \beta_2y + \gamma_2}{\alpha_3x + \beta_3y + \gamma_3}\right). \end{array}$$

Les déformations φ et ψ liant les images f et g sont donc des applications projectives, respectivement associées aux matrices $\mathcal{M}_\varphi$ et $\mathcal{M}_\psi$

$$\mathcal{M}_\varphi = \begin{pmatrix} a_1 & b_1 & c_1 + \widetilde{t_1} \\ a_2 & b_2 & c_2 + \widetilde{t_2} \\ a_3 & b_3 & c_3 + \widetilde{t_3} \end{pmatrix}$$

et

$$\mathcal{M}_\psi = \begin{pmatrix} a_1 & a_2 & a_3 - \langle \widetilde{t}, R(i) \rangle \\ b_1 & b_2 & b_3 - \langle \widetilde{t}, R(j) \rangle \\ c_1 & c_2 & c_3 - \langle \widetilde{t}, R(k) \rangle \end{pmatrix}.$$

Proposition 2.2 (Dibos 2001 [11]) – *Soient $D = (R, t)$ un mouvement de caméra, f et g les images acquises avant et après le déplacement et φ et ψ les applications projectives liant les deux images. Alors, la matrice $\mathcal{M}_\varphi$ s'écrit de façon unique comme produit de la matrice de rotation R et d'une matrice H dépendant de la translation [12]*

$$\mathcal{M}_\varphi = R \begin{pmatrix} 1 & 0 & \langle \widetilde{t}, R(i) \rangle \\ 0 & 1 & \langle \widetilde{t}, R(j) \rangle \\ 0 & 0 & 1 + \langle \widetilde{t}, R(k) \rangle \end{pmatrix} = RH. \tag{2.7}$$

Réciproquement, la matrice $\mathcal{M}_\psi$ se décompose de façon unique en la matrice de rotation R^{-1} et une matrice $\widetilde{H}$ dépendant de la translation

$$\mathcal{M}_\psi = R^{-1} \begin{pmatrix} 1 & 0 & -\widetilde{t_1} \\ 0 & 1 & -\widetilde{t_2} \\ 0 & 0 & 1 - \widetilde{t_3} \end{pmatrix} = R^{-1} \widetilde{H}.$$

Démonstration. On vérifie que le produit RH est bien égal à $\mathcal{M}_\varphi$. De plus, s'il existait H_1 et H_2 telles que $\mathcal{M}_\varphi = RH_1 = RH_2$, on aurait alors $R(H_1 - H_2) = 0$; comme R est inversible, on obtient que $H_1 = H_2$, ce qui prouve l'unicité de la décomposition. □

Remarque – L'application φ est une application projective si $\det(\mathcal{M}_\varphi)$ est non nul. Or,

$$\det(\mathcal{M}_\varphi) = \det(RH) = \det(H) = 1 + \langle \widetilde{t}, R(k) \rangle.$$

Le cas $\det(\mathcal{M}_\varphi) = 0$ ne se produira pas dans le contexte auquel nous nous intéressons car $\langle \widetilde{t}, R(k) \rangle = -1$ équivaut à $\langle t, R(k) \rangle = -Z_0$ ce qui contredit l'hypothèse que

$$\frac{\|t\|}{Z_0} (L + 1) G_{max}$$

est petit ; on ne peut pas avoir une composante de la translation de la caméra entre deux images consécutives égale à la profondeur de la scène. De la même façon, le cas $\det(\mathcal{M}_\psi) = 0$ ne se présentera pas car il faudrait alors que t_3 soit égal à Z_0.

2.2.2 Groupe projectif

L'ensemble des applications projectives, muni de la multiplication matricielle, forme un groupe car le produit de deux matrices inversibles est inversible.

Définition 2.3 – *L'ensemble des applications projectives de* $\mathbb{R}^2$ *dans* $\mathbb{R}^2$ *est appelé groupe projectif et noté* $GP^2(\mathbb{R})$.

D'après ce qui précède,

$$GP^2(\mathbb{R}) = \Big\{ \phi : \mathbb{R}^2 \to \mathbb{R}^2 \text{ telles que } \forall (x,y) \in \mathbb{R}^2,$$

$$\phi(x,y) = \left(\frac{\alpha_1 x + \beta_1 y + \gamma_1}{\alpha_3 x + \beta_3 y + \gamma_3}, \frac{\alpha_2 x + \beta_2 y + \gamma_2}{\alpha_3 x + \beta_3 y + \gamma_3} \right) \text{ avec } \det \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix} \neq 0 \Big\}.$$

Ce groupe est aussi l'ensemble des homographies du plan rétinien $\mathcal{R}$ dans lui-même, c'est-à-dire l'ensemble des transformations de $\mathcal{R}$ dans $\mathcal{R}$ linéaires en coordonnées projectives et inversibles.

Proposition 2.3 – *Le groupe projectif* $GP^2(\mathbb{R})$ *est isomorphe au groupe Spécial Linéaire* $SL(3,\mathbb{R})$.

Démonstration. Le groupe projectif est isomorphe au groupe des matrices inversibles et de déterminant égal à un, car l'application générée par une matrice A est identique à celle générée par la matrice λA, pour tout réel λ. □

Remarques

- En conséquence de la proposition, les groupes $SL(3,\mathbb{R})$ et $GP^2(\mathbb{R})$ dépendent du même nombre de paramètres, c'est-à-dire 8.
- Les transformations φ et ψ étant des applications projectives, elles sont souvent représentées dans le groupe projectif [15].

Nous allons maintenant mettre en évidence un lien entre le groupe projectif et la décomposition des matrices $\mathcal{M}_\varphi = RH$ et $\mathcal{M}_\psi = R^{-1}\widetilde{H}$, décomposition qui fait apparaître le produit d'une matrice de rotation avec une matrice associée à la translation.

Définition 2.4 – *On appelle* $\mathcal{T}$ *le groupe des matrices de* $\mathcal{M}_3(\mathbb{R})$ *de la forme*

$$\frac{1}{C} \begin{pmatrix} 1 & 0 & A \\ 0 & 1 & B \\ 0 & 0 & C^3 \end{pmatrix} \quad \textit{avec } C \neq 0.$$

À une normalisation près pour que leur déterminant soit égal à un, les matrices H, mentionnées dans la décomposition (2.7) de la matrice $\mathcal{M}_\varphi$, appartiennent au groupe $\mathcal{T}$. Nous allons montrer le théorème suivant

Théorème 2.2 – *Le groupe spécial linéaire $SL(3,\mathbb{R})$ est engendré par $SO(3) \cup \mathcal{T}$, où $SO(3)$ est le groupe spécial orthogonal.*

Démonstration. Commençons par le lemme :

Lemme 2.1 – *Le groupe spécial linéaire $SL(3,\mathbb{R})$ est engendré par l'ensemble des transvections dans $\mathbb{R}^3$, c'est-à-dire par l'ensemble des matrices de la forme $A_{ij} = I_3 + \alpha E_{ij}$ où $\alpha \in \mathbb{R}$, $i,j \in \{1,2,3\}$, $i \neq j$, et E_{ij} est la matrice dont tous les coefficients sont nuls sauf celui de la ligne i et de la colonne j qui vaut 1.*

En effet, toute matrice de $SL(3,\mathbb{R})$ peut s'écrire comme produit de matrices de transvection. Nous ne développons pas la preuve mais l'idée est la suivante : soit M une matrice de $SL(3,\mathbb{R})$, on peut la multiplier par des matrices de transvection de sorte que le terme en haut à gauche soit égal à 1 (en normalisant) et soit le seul de la ligne à être non nul. On procède ensuite par récurrence sur les lignes de M. L'inverse d'une matrice de transvection étant une transvection, on obtient le résultat.

Ainsi, pour montrer que $SL(3,\mathbb{R})$ est engendré par $SO(3) \cup \mathcal{T}$, il suffit de montrer que l'ensemble des transvections est engendré par $SO(3) \cup \mathcal{T}$. Soient $\alpha \in \mathbb{R}$, $i,j \in \{1,2,3\}$ et $i \neq j$. Nous allons montrer que les six matrices de transvection $A_{ij} = I_3 + \alpha E_{ij}$ peuvent être obtenues en multipliant des matrices du groupe $\mathcal{T}$ avec les deux matrices de rotations

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \text{ et } \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

Tout d'abord, les matrices

$$A_{13} = \begin{pmatrix} 1 & 0 & \alpha \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } A_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha \\ 0 & 0 & 1 \end{pmatrix}$$

sont des matrices du groupe $\mathcal{T}$. Ensuite, les matrices A_{31} et A_{32} sont obtenues par multiplication de deux matrices du groupe $\mathcal{T}$ avec une matrice de rotation

$$A_{31} = \underbrace{\frac{1}{-\alpha^{1/3}} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & -\alpha \end{pmatrix}}_{\in \mathcal{T}} \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}}_{\in SO(3)} \underbrace{(-\alpha^{1/3}) \begin{pmatrix} 1 & 0 & 1/\alpha \\ 0 & 1 & 0 \\ 0 & 0 & -1/\alpha \end{pmatrix}}_{\in \mathcal{T}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & 0 & 1 \end{pmatrix}$$

et

$$A_{32} = \underbrace{\frac{1}{\alpha^{1/3}}\begin{pmatrix}1&0&0\\0&1&1\\0&0&\alpha\end{pmatrix}}_{\in\mathcal{T}}\underbrace{\begin{pmatrix}1&0&0\\0&0&-1\\0&1&0\end{pmatrix}}_{\in SO(3)}\underbrace{(\alpha^{1/3})\begin{pmatrix}1&0&0\\0&1&1/\alpha\\0&0&1/\alpha\end{pmatrix}}_{\in\mathcal{T}} = \begin{pmatrix}1&0&0\\0&1&0\\0&\alpha&1\end{pmatrix}.$$

Enfin, les deux matrices A_{12} et A_{21} sont obtenues par multiplication de matrices de transvections du type A_{32} ou A_{31} avec deux matrices du groupe $\mathcal{T}$

$$A_{12} = \underbrace{\begin{pmatrix}1&0&-1\\0&1&0\\0&0&1\end{pmatrix}}_{\in\mathcal{T}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\0&-\alpha&1\end{pmatrix}}_{\text{matrice du type } A_{32}}\underbrace{\begin{pmatrix}1&0&1\\0&1&0\\0&0&1\end{pmatrix}}_{\in\mathcal{T}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\0&\alpha&1\end{pmatrix}}_{\text{matrice du type } A_{32}}$$

et

$$A_{21} = \underbrace{\begin{pmatrix}1&0&0\\0&1&\alpha\\0&0&1\end{pmatrix}}_{\in\mathcal{T}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\1&0&1\end{pmatrix}}_{\text{matrice du type } A_{31}}\underbrace{\begin{pmatrix}1&0&0\\0&1&-\alpha\\0&0&1\end{pmatrix}}_{\in\mathcal{T}}\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\-1&0&1\end{pmatrix}}_{\text{matrice du type } A_{31}}.$$

Ainsi, toutes les matrices de transvections sont obtenues par multiplication de matrices de rotations et de matrices du groupe $\mathcal{T}$. Donc le groupe $SL(3,\mathbb{R})$, engendré par les transvections, est aussi engendré par $SO(3) \cup \mathcal{T}$. □

Comme le groupe projectif est isomorphe au groupe spécial linéaire, cela revient à dire que le groupe projectif est engendré par $SO(3) \cup \mathcal{T}$; ceci relie les décompositions $\mathcal{M}_\varphi = RH$ et $\mathcal{M}_\psi = R^{-1}\widetilde{H}$ au groupe projectif.

2.2.3 Observations

La représentation des applications projectives φ et ψ dans le groupe projectif présente plusieurs inconvénients. D'une part, tout produit RH, $R \in SO(3)$ et $H \in \mathcal{T}$, peut toujours être associé à un mouvement de caméra mais ce n'est pas vrai pour tout produit HR. Ceci est dû au fait que le groupe projectif est défini par huit paramètres et un mouvement de caméra par seulement six. D'autre part, dans le groupe projectif, l'application ψ, qui permet de retrouver l'image f à partir de g n'est pas égale à l'application φ^{-1}, c'est-à-dire que la matrice $\mathcal{M}_\psi$ est différente de $\mathcal{M}_\varphi^{-1}$ dans le groupe $SL(3,\mathbb{R})$.

Considérons les images f, g et h d'une même scène obtenues par déplacements successifs de la caméra. Les images f, g et h ont été acquises respectivement, avant le déplacement D_1 de la caméra, après D_1 et avant D_2, et après D_2. Soient (C,i,j,k), (C',i',j',k') et (C'',i'',j'',k'') les repères orthonormés associés aux positions et orientations successives de la caméra. On sait que

$$g = f \circ \varphi_1 \quad \text{et} \quad h = g \circ \varphi_2,$$

où les matrices de φ_1 et φ_2 sont obtenues respectivement grâce à l'écriture $D_1 = (R_1, t_1)$ dans le repère (C, i, j, k) et $D_2 = (R_2, t_2)$ dans le repère (C', i', j', k'). On cherche à déterminer la loi de composition $\varphi_1 \star \varphi_2$ permettant d'avoir

$$h = f \circ (\varphi_1 \star \varphi_2).$$

Il suffit pour cela d'écrire la matrice de rotation et le vecteur translation du déplacement de la caméra entre les acquisitions de f et h, dans le repère (C, i, j, k). L'écriture de R_2 dans (C, i, j, k) est la matrice

$$R_1 R_2 R_1^{-1},$$

par conséquent, la composition des rotations dans (C, i, j, k) est donnée par

$$(R_1 R_2 R_1^{-1}) R_1 = R_1 R_2.$$

De même, l'écriture de t_2 dans (C, i, j, k) est le vecteur

$$R_1 t_2,$$

donc le vecteur translation CC'' s'écrit dans (C, i, j, k)

$$t_1 + R_1 t_2.$$

Finalement, la matrice associée à $\varphi_1 \star \varphi_2$ s'écrit

$$\mathcal{M}_{\varphi_1 \star \varphi_2} = R_1 R_2 \begin{pmatrix} 1 & 0 & \langle \tilde{t}_1 + R_1 \tilde{t}_2, R_1 R_2(i) \rangle \\ 0 & 1 & \langle \tilde{t}_1 + R_1 \tilde{t}_2, R_1 R_2(j) \rangle \\ 0 & 0 & 1 + \langle \tilde{t}_1 + R_1 \tilde{t}_2, R_1 R_2(k) \rangle \end{pmatrix}$$

où $\tilde{t}_1$ et $\tilde{t}_2$ sont les translations de la caméra divisées par la profondeur Z_0 de la scène. Nous allons ainsi choisir un nouveau groupe pour modéliser φ et ψ, dans lequel une déformation entre deux images consécutives correspondra à un mouvement de caméra.

2.2.4 Groupe des recalages

Le groupe des recalages provient de la modélisation d'un mouvement de caméra dans le groupe $SE(3)$ des déplacements rigides de $\mathbb{R}^3$.

Définition 2.5 (Dibos 2001 [11]) – *Soit $\mathcal{A}$ l'ensemble des fonctions $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ telles que*

$$\forall (x, y) \in \mathbb{R}^2, \quad \phi(x, y) = \left(\frac{a_1 x + b_1 y + c_1 + \alpha}{a_3 x + b_3 y + c_3 + \gamma}, \frac{a_2 x + b_2 y + c_2 + \beta}{a_3 x + b_3 y + c_3 + \gamma} \right),$$

où

$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \in SO(3) \quad \textit{et} \quad (\alpha, \beta, \gamma) \in \mathbb{R}^3.$$

On appelle groupe des recalages le groupe $(\mathcal{A}, \star)$ *isomorphe au groupe* $(SE(3), \circ)$ *des déplacements rigides de* $\mathbb{R}^3$. *L'isomorphisme de groupes* $\mathcal{I}$ *est défini par*

$$\mathcal{I} : \mathcal{A} \longrightarrow SE(3)$$

$$\forall \phi \in \mathcal{A} \quad \mathcal{I}(\phi) = (R, t)$$

où R *est la rotation définie ci-avant et* t *est la translation de vecteur* (α, β, γ).

Plus précisément, si ϕ_1 et ϕ_2 sont deux applications du groupe $\mathcal{A}$, elles correspondent respectivement aux déplacements $D_1 = (R_1, t_1)$ et $D_2 = (R_2, t_2)$. On a alors $\phi_1 \star \phi_2 = \phi$ où ϕ est l'application projective du groupe $\mathcal{A}$ associée au déplacement $D = D_1 \circ D_2 = (R, t)$ avec $R = R_1 R_2$ et t la translation de vecteur $t_1 + R_1 t_2$ (car $SE(3) = SO(3) \ltimes \mathbb{R}^3$).

Remarquons que dans le groupe des recalages, si l'application φ est associée au mouvement $D = (R, t)$ alors ψ est associée au mouvement $D^{-1} = (R^{-1}, -R^{-1}t)$. Ceci signifie que dans le groupe des recalages, l'application ψ est l'inverse de φ,

$$\psi = \varphi^{-1}.$$

Notons qu'il est essentiel pour nous de modéliser les déformations d'images dans un groupe. En effet, la structure de groupe permet d'inverser et de composer des déformations (inversions et compositions étant associées à celles des déplacements de caméra). À partir des déplacements estimés entre des images consécutives d'une séquence, on peut en déduire, en combinant les faibles déplacements, le mouvement de la caméra entre des images éloignées dans le temps. Pour la translation, ceci n'est possible que si la scène filmée reste suffisamment éloignée de la caméra tout au long du mouvement, tandis que l'on aura dans tous les cas une estimation de la rotation.

Plus formellement, soient $f_1,\ f_2,\ f_3 \ldots f_n$ une suite d'images d'une même scène, obtenues par déplacements successifs de la caméra. À chaque étape, on connaît l'application projective φ_i telle que

$$f_{i+1} = f_i \circ \varphi_i$$

et donc aussi le déplacement $D_i = (R_i, t_i)$. Alors, si la scène demeure suffisamment éloignée de la caméra pendant les n acquisitions d'image,

$$f_n = f_1 \circ (\varphi_1 \star \varphi_2 \star \ldots \star \varphi_{n-1}) = f_1 \circ \varphi$$

où φ est l'application projective associée à la matrice de rotation

$$R = R_1 R_2 \ldots R_{n-1}$$

et à la translation

$$t = t_1 + R_1 t_2 + R_1 R_2 t_3 + \ldots + R_1 R_2 \ldots R_{n-2} t_{n-1}.$$

Remarques

- Entre deux acquisitions d'images consécutives, la direction de l'axe optique est très peu modifiée ; le vecteur $R(k)$ demeure très proche du vecteur k, ce qui implique que $c_1 \approx 0$ et $c_2 \approx 0$. Les déformations affines sont donc de "bonnes" approximations des déformations projectives observées entre des images consécutives dans la séquence. Ceci explique le rôle important que le groupe affine a joué [59, 10].
- Cependant, les déformations affines exactes générées par un mouvement de caméra ne peuvent être que des similitudes. En effet, si la déformation ψ est affine, $c_1 = c_2 = 0$ et comme R est une matrice orthogonale positive, $a_3 = b_3 = 0$ et $c_3 = 1$. Ainsi, R est une rotation d'axe k.

2.3 Décomposition d'un mouvement de caméra

Dans cette section, on propose une décomposition non standard d'un mouvement de caméra, permettant de séparer la déformation produite entre les deux images consécutives en deux composantes : une similitude et une déformation "purement" projective. En effet, on peut toujours décomposer un déplacement de caméra en trois mouvements de base :

- une translation, qui produit une homothétie-translation sur l'image f du plan rétinien $\mathcal{R}$,
- une rotation d'axe k, qui produit une rotation plane de l'image f,
- une rotation d'axe appartenant au plan (C, i, j), qui entraîne une déformation projective de l'image f.

2.3.1 Décomposition d'une rotation

Considérons une rotation de caméra R d'axe contenant le centre optique C. On décompose R en deux rotations particulières $R_2 R_1$.

La première, R_1, d'axe Δ appartenant au plan (C, i, j), modifie la direction de l'axe optique k. L'axe Δ et l'angle de rotation sont choisis de telle sorte qu'après la rotation R_1, l'axe pointe dans la direction $R(k)$. Du point de vue de l'image, en considérant la caméra immobile, cette rotation provoque une déformation de l'image f que nous qualifierons de "purement" projective. La seconde rotation, R_2, est une rotation autour de l'axe $R(k)$. Après cette seconde rotation, les axes $R_1(i)$ et $R_1(j)$ sont transformés en $R(i)$ et $R(j)$. La rotation R_2 induit une rotation plane de l'image f déformée par R_1. Cette décomposition est représentée sur la figure (2.4).

Formellement, la rotation R_1 dépend de deux paramètres, notés θ et α ; l'angle θ localise l'axe Δ dans le plan (C, i, j) et l'angle α est l'angle de rotation autour de Δ. Si on note R_a^u la matrice de rotation d'axe u et d'angle a, l'expression de R_1 dans le repère (C, i, j, k) est la suivante

$$R_1 = R_\theta^k R_\alpha^i R_{-\theta}^k$$

et nous noterons dans la suite la rotation R_1 par $R_{\theta,\alpha}$. La rotation R_2 ne dépend que d'un paramètre : son angle de rotation β autour du nouvel axe optique (après la rotation $R_{\theta,\alpha}$). Dans

le repère (C, i, j, k), l'écriture de la rotation R_2 est la suivante

$$R_2 = R_{\theta,\alpha} R^k_\beta R_{\theta,-\alpha} = R^k_\theta R^i_\alpha R^k_\beta R^i_{-\alpha} R^k_{-\theta}.$$

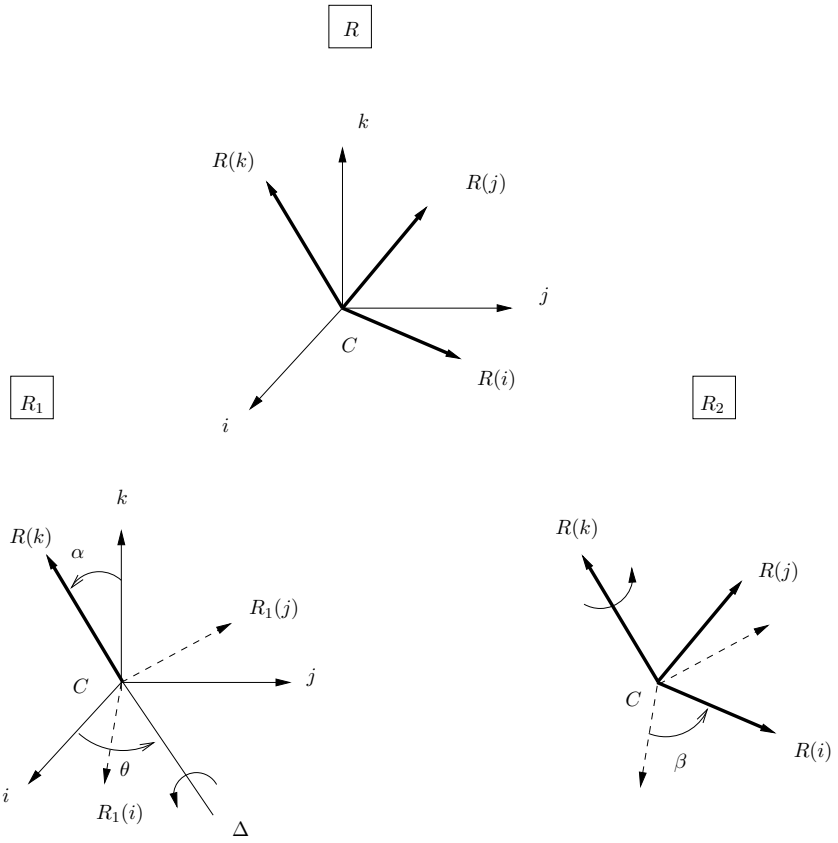

FIGURE 2.4: *Décomposition d'une rotation de caméra R en deux rotations $R_2 R_1$.*

En définitive, l'expression de la rotation complète R dans le repère (C, i, j, k) est

$$R = R_2 R_1 = R^k_\theta R^i_\alpha R^k_\beta R^k_{-\theta}.$$

Cette décomposition est intéressante en regard des déformations générées par chacune de ses composantes ; R_1 produit d'abord une déformation "purement" projective de l'image f tandis que R_2 entraîne une rotation plane de l'image déformée par R_1.

La décomposition de la déformation générée par la rotation de la caméra sur l'image est illustrée sur la figure (2.5). L'image de Lena est déformée par une rotation de caméra paramétrée par $\theta = 0$, $\alpha = 0.1$ et $\beta = 0.2$ (en radians). Cette image correspond bien à l'image de Lena déformée par la transformation "purement projective" paramétrée par les angles $\theta = 0$ et $\alpha = 0.1$, à laquelle on fait subir une rotation plane d'angle $\beta = 0.2$. Les déformations sont appliquées

aux images en utilisant une interpolation bilinéaire. Cet exemple utilise des valeurs d'angles supérieures à celles correspondant aux déformations observées entre deux images consécutives dans une séquence (qui seront précisées dans le tableau (2.1)), afin de rendre plus évidente l'observation de la décomposition sur les images. Pour cet exemple, nous avons choisi un angle de vue égal à 137°, ce qui revient à prendre une image de taille 5.12×5.12 en unités de longueur focale.

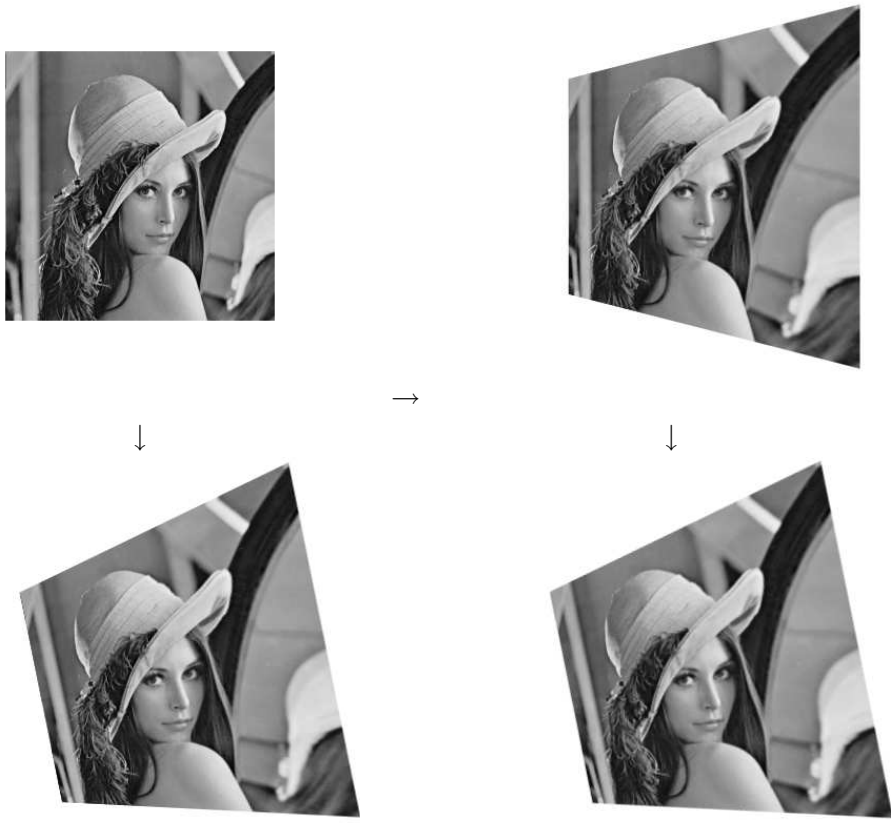

FIGURE 2.5: *Illustration de la décomposition d'une rotation de caméra. À gauche, de haut en bas, l'image de Lena et l'image obtenue par une rotation paramétrée par* $\theta = 0,\ \alpha = 0.1,\ \beta = 0.2$. *À droite, de haut en bas, l'image de Lena filmée par la caméra après une rotation* $R_1 = R_{\theta,\alpha}$ *avec* $\theta = 0,\ \alpha = 0.1$ *puis cette image après une rotation* R_2 *de paramètre* $\beta = 0.2$. *L'image initiale est de taille* 5.12×5.12 *en unités de longueur focale.*

À partir des trois angles θ, α et β, on peut exprimer l'axe de la rotation de la caméra et l'angle de la rotation. Écrivons pour cela l'expression de la matrice R en fonction de θ, α et β.

Dans l'écriture précédente de R, on peut permuter R^k_β et $R^k_{-\theta}$ car ces deux rotations ont même axe donc

$$R = R_1 R^k_\beta = R_{\theta,\alpha} R^k_\beta.$$

Les matrices $R_{\theta,\alpha}$ et R^k_β s'écrivent

$$R_{\theta,\alpha} = \begin{pmatrix} \cos^2\theta + \sin^2\theta\cos\alpha & \cos\theta\sin\theta(1-\cos\alpha) & \sin\theta\sin\alpha \\ \cos\theta\sin\theta(1-\cos\alpha) & \sin^2\theta + \cos^2\theta\cos\alpha & -\cos\theta\sin\alpha \\ -\sin\theta\sin\alpha & \cos\theta\sin\alpha & \cos\alpha \end{pmatrix},$$

$$R^k_\beta = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

soit

$$R = \begin{pmatrix} \cos\beta - (1-\cos\alpha)\sin\theta\sin(\theta-\beta) & -\sin\beta + (1-\cos\alpha)\sin\theta\cos(\theta-\beta) & \sin\theta\sin\alpha \\ \sin\beta + (1-\cos\alpha)\cos\theta\sin(\theta-\beta) & \cos\beta - (1-\cos\alpha)\cos\theta\cos(\theta-\beta) & -\cos\theta\sin\alpha \\ -\sin\alpha\sin(\theta-\beta) & \sin\alpha\cos(\theta-\beta) & \cos\alpha \end{pmatrix}. \quad (2.8)$$

Comme $R - R^T = 2\sin a\,[u]_\times$ où u est un vecteur directeur unitaire de l'axe de la rotation et a son angle, on obtient que l'axe de rotation de R est dirigé suivant le vecteur u, non unitaire

$$\begin{pmatrix} \sin\alpha\,(\cos\theta + \cos(\theta-\beta)) \\ \sin\alpha\,(\sin\theta + \sin(\theta-\beta)) \\ \sin\beta\,(1+\cos\alpha) \end{pmatrix}.$$

On a aussi $tr(R) = 2\cos a + 1$; ainsi, l'angle a de la rotation ($a \geq 0$) est égal à

$$\arccos\left(\frac{\cos\alpha + \cos\beta + \cos\alpha\cos\beta - 1}{2}\right)$$

ou, comme a est compris entre 0 et $\pi/2$ (pour une rotation de caméra entre deux acquisitions consécutives d'images),

$$\arcsin\sqrt{\frac{1}{4}\sin^2\beta\,(1+\cos\alpha)^2 + \frac{1}{2}\sin^2\alpha(1+\cos\beta)}.$$

Inversement, à partir d'une matrice de rotation $R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$, on peut obtenir ($\theta$, α, β) par

– si $c_3 \neq 1$

$$\alpha = \arccos(c_3)$$

$$\theta = \begin{cases} \arctan(-c_1/c_2) & \text{si } c_2 < 0 \\ \arctan(-c_1/c_2) + \pi & \text{si } c_2 > 0 \\ \pi/2 & \text{si } c_1 > 0 \text{ et } c_2 = 0 \\ \pi/2 & \text{si } c_1 < 0 \text{ et } c_2 = 0 \end{cases}$$

$$\beta = \begin{cases} \theta + \arctan(a_3/b_3) & \text{si } b_3 > 0 \\ \theta + \arctan(a_3/b_3) + \pi & \text{si } b_3 < 0 \\ \theta + \pi/2 & \text{si } a_3 > 0 \text{ et } b_3 = 0 \\ \theta - \pi/2 & \text{si } a_3 < 0 \text{ et } b_3 = 0, \end{cases}$$

– si $c_3 = 1$

$$\alpha = \theta = 0$$

$$\beta = \begin{cases} \arctan(-a_2/a_1) & \text{si } a_1 > 0 \\ \arctan(-a_2/a_1) + \pi & \text{si } a_1 < 0 \\ \pi/2 & \text{si } a_2 > 0 \text{ et } a_1 = 0 \\ \pi/2 & \text{si } a_2 < 0 \text{ et } a_1 = 0. \end{cases}$$

2.3.2 Décomposition d'un mouvement complet

Un mouvement de caméra complet $D = (R, t)$ produit une déformation projective φ sur l'image f. La matrice associée à φ, décrite dans la formule (2.7) peut maintenant s'écrire,

$$\mathcal{M}_\varphi = RH = R_{\theta,\alpha} R^k_\beta H.$$

Si on note $r_{\theta,\alpha}$ l'application "purement" projective associée à $R_{\theta,\alpha}$ et s la similitude associée à $R^k_\beta H$, on a

$$g(x, y) = f(\varphi(x, y)) = f(r_{\theta,\alpha} \circ s(x, y)) = (f \circ r_{\theta,\alpha})(s(x, y)).$$

La déformation φ de l'image f revient donc à transformer l'image par l'application projective $r_{\theta,\alpha}$ puis à appliquer à cette nouvelle image la similitude s.

Les six paramètres définissant le mouvement d'une caméra sont alors répartis comme suit : deux paramètres pour la rotation $R_{\theta,\alpha}$ et quatre paramètres pour la translation $\widetilde{t}$ et la rotation R^k_β, soit deux pour l'application "purement" projective et quatre pour la similitude observée sur l'image. On exprime dorénavant un mouvement de caméra par les six paramètres suivants $(\theta, \alpha, \beta, A, B, C)$ où $(-A, -B, -C)$ sont les coordonnées de la translation $\widetilde{t}$ dans la base $(R(i), R(j), R(k))$. Ces nouvelles notations simplifient l'écriture de l'application projective ψ, inverse de φ dans le groupe des recalages, et que nous utiliserons plus tard pour le calcul du flot optique.

Proposition 2.4 – *Dans le cas d'une scène initiale plane et orthogonale à l'axe optique de la caméra, tout mouvement de caméra $D = (R, t) \in SE(3)$ peut s'écrire $D = (\theta, \alpha, \beta, A, B, C)$ où (θ, α, β) définissent la rotation et $(A, B, C) = -(\langle t/Z_0, R(i)\rangle, \langle t/Z_0, R(j)\rangle, \langle t/Z_0, R(k)\rangle)$, où Z_0 est la profondeur de la scène dans le repère associé à la caméra avant le déplacement.*

Dans le cas où $f_c = 1$, K et K' étant les domaines sur lesquels sont définies f et g, un point (x, y) de K est apparié à un point (x', y') de K' par

$$(x', y') = \psi(x, y) = \left(\frac{a_1 x + a_2 y + a_3 + A}{c_1 x + c_2 y + c_3 + C}, \frac{b_1 x + b_2 y + b_3 + B}{c_1 x + c_2 y + c_3 + C} \right). \tag{2.9}$$

Remarquons que les six paramètres $(\theta, \alpha, \beta, A, B, C)$ permettent d'accéder facilement au mouvement de la caméra. En effet,

$$\begin{cases} \widetilde{t} = -AR(i) - BR(j) - CR(k) \\ \\ R = R_{\theta,\alpha} R_{\beta}^{k}. \end{cases} \tag{2.10}$$

La décomposition de la déformation générée par le mouvement de la caméra est illustrée sur la figure (2.6). À l'image de Lena, on applique la déformation générée par un mouvement de caméra paramétré par $(\theta, \alpha, \beta, A, B, C) = (-\pi/2, 0.1, -0.2, 0.5, -0.5, -0.05)$, les angles étant exprimés en radians. Cette déformation revient à appliquer à l'image la déformation "purement" projective $r_{\theta,\alpha}$ avec $\theta = -\pi/2$ et $\alpha = 0.1$, puis la similitude s correspondant à la rotation d'angle $\beta = -0.2$ suivie de l'homothétie-translation de paramètres $(A, B, C) = (0.5, -0.5, -0.05)$. Ici encore, le mouvement de caméra choisi est plus important que celui observé entre deux images consécutives d'une séquence. Comme pour l'illustration de la décomposition de la rotation, on a choisi l'angle de vue de la caméra égal à 137°, ce qui revient à prendre une image de taille 5.12×5.12 en unités de longueur focale.

Cette décomposition du mouvement de la caméra est intéressante car elle correspond à notre perception des effets du mouvement entre deux images. L'oeil différencie aisément les déformations dues à la modification de la direction de l'axe optique, qui ne préservent pas par exemple les lignes parallèles de l'image, des effets de la similitude, qui conservent les angles et les rapports des distances.

2.4 Approximation et décomposition du flot optique

Nous allons maintenant nous intéresser au flot optique entre deux images consécutives dans une séquence en fonction des six paramètres définis ci-avant, d'abord dans le cas d'une scène plane et orthogonale à l'axe optique avant le déplacement puis dans le cas général. Auparavant, nous rappelons le rôle de la longueur focale dans les expressions des déformations.

FIGURE 2.6: *Illustration de la décomposition d'un mouvement complet de caméra. À gauche, de haut en bas, l'image de Lena, et l'image filmée par la caméra après un mouvement paramétré par* $(\theta, \alpha, \beta, A, B, C) = (-\pi/2, 0.1, -0.2, 0.5, -0.5, -0.05)$. *À droite, de haut en bas, l'image obtenue après une rotation* $R_1 = R_{\theta,\alpha}$ *avec* $\theta = -\pi/2,\ \alpha = 0.1$ *puis après l'application de la similitude de paramètres* $(\beta, A, B, C) = (-0.2, 0.5, -0.5, -0.05)$. *L'image initiale est de taille* 5.12×5.12 *en unités de longueur focale.*

2.4.1 Rôle de la longueur focale

Dans ce qui précède, nous avons supposé la longueur focale f_c égale à 1, ce qui revient à la considérer comme unité du repère associé à la caméra (C, i, j, k) et du repère de l'image (c, i, j). Que se passe-t-il si la longueur focale n'est plus égale à 1 ? Soient $\mathcal{R}$ et $\widetilde{\mathcal{R}}$ les plans rétiniens situés à des longueurs focales respectives $f_c = 1$ et $f_c \neq 1$ de la caméra et de repères respectifs (c, i, j) et $(\tilde{c}, i, j)$, comme illustré sur la figure (2.7). Soit un point de l'espace de coordonnées (X, Y, Z) dans (C, i, j, k) et ses projections (x, y) et $(\tilde{x}, \tilde{y})$ sur $\mathcal{R}$ et $\widetilde{\mathcal{R}}$. Ces projections sont liées

par

$$\begin{cases} \tilde{x} = f_c \frac{X}{Z} = f_c\, x \\ \\ \tilde{y} = f_c \frac{Y}{Z} = f_c\, y. \end{cases}$$

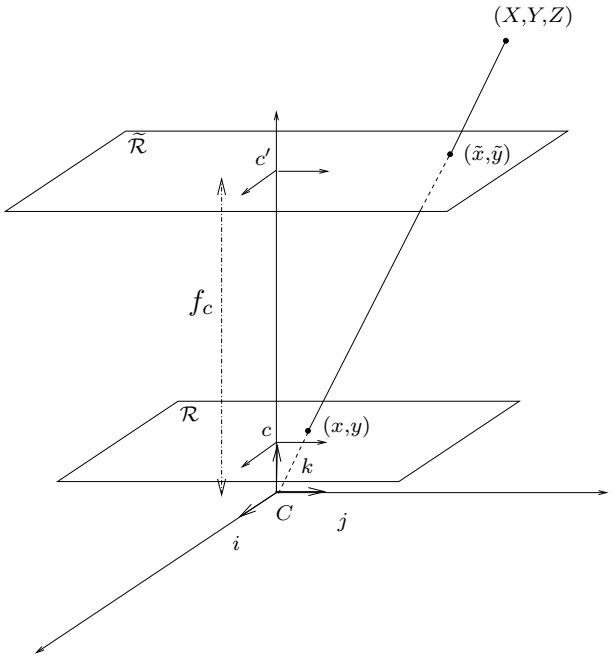

FIGURE 2.7: *Rôle de la longueur focale.*

Considérons maintenant un déplacement de la caméra de rotation

$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$$

et de translation

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}.$$

Soit un point de $\mathbb{R}^3$, de profondeur Z dans (C, i, j, k), projeté en (x, y) et $(\tilde{x}, \tilde{y})$ sur $\mathcal{R}$ et $\widetilde{\mathcal{R}}$ avant le déplacement et en (x', y') et $(\widetilde{x'}, \widetilde{y'})$ sur $\mathcal{R}$ et $\widetilde{\mathcal{R}}$ après le déplacement. Reprenons l'écriture de

l'application projective ψ liant (x', y') à (x, y)

$$\begin{cases} x' = \dfrac{a_1 x + a_2 y + a_3 - \langle \frac{t}{Z}, R(i) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z}, R(k) \rangle} \\ \\ y' = \dfrac{b_1 x + b_2 y + b_3 - \langle \frac{t}{Z}, R(j) \rangle}{c_1 x + c_2 y + c_3 - \langle \frac{t}{Z}, R(k) \rangle}. \end{cases}$$

Comme $(\tilde{x}, \tilde{y}) = (f_c\, x, f_c\, y)$ et $(\widetilde{x'}, \widetilde{y'}) = (f_c\, x', f_c\, y')$, on obtient

$$\begin{cases} \dfrac{\widetilde{x'}}{f_c} = \dfrac{a_1 \dfrac{\tilde{x}}{f_c} + a_2 \dfrac{\tilde{y}}{f_c} + a_3 - \langle \dfrac{t}{Z}, R(i) \rangle}{c_1 \dfrac{\tilde{x}}{f_c} + c_2 \dfrac{\tilde{y}}{f_c} + c_3 - \langle \dfrac{t}{Z}, R(k) \rangle} \\ \\ \dfrac{\widetilde{y'}}{f_c} = \dfrac{b_1 \dfrac{\tilde{x}}{f_c} + b_2 \dfrac{\tilde{y}}{f_c} + b_3 - \langle \dfrac{t}{Z}, R(j) \rangle}{c_1 \dfrac{\tilde{x}}{f_c} + c_2 \dfrac{\tilde{y}}{f_c} + c_3 - \langle \dfrac{t}{Z}, R(k) \rangle}. \end{cases}$$

Ainsi, le choix de la longueur focale égale à 1 ne modifie que l'échelle d'observation des images comme nous l'avions vu dans le chapitre 1 et l'échelle des déformations d'images générées par un mouvement de caméra.

2.4.2 Approximation et décomposition du flot optique

Exprimons maintenant le flot optique, c'est-à-dire le déplacement des points entre deux images consécutives d'une séquence, en fonction des valeurs des six paramètres définis précédemment.

Théorème 2.3 – *Soient $D = (R, t) \in SE(3)$, $t \neq 0$, $f_c = 1$ et K et K' les domaines de plus grande dimension L sur lesquels les images f et g sont définies. On suppose la scène plane et orthogonale à l'axe optique avant le déplacement : on note $D = (\theta, \alpha, \beta, A, B, C)$. On suppose que D et Z vérifient l'hypothèse (1), que $|\alpha| < 1$ et $|\beta| < 1$. Soient $(x, y) \in K$ et $(x', y') = \psi(x, y)$. Alors, le flot optique au point (x, y) vérifie*

$$\begin{cases} x' - x = & -Cx + A + \beta y + \alpha x(y\cos\theta - x\sin\theta) - \alpha\sin\theta + o(C) + o(\alpha) + o(\beta) \\ & +o(\sqrt{|\alpha A|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|AC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}) \\ y' - y = & -Cy + B - \beta x + \alpha y(y\cos\theta - x\sin\theta) + \alpha\cos\theta + o(C) + o(\alpha) + o(\beta) \\ & +o(\sqrt{|\alpha B|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|BC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}) \end{cases}$$

et

$$\begin{cases} \left|x' - x - (-Cx + A + \beta y + \alpha x(y\cos\theta - x\sin\theta) - \alpha\sin\theta)\right| \leq T(G_{max}, L, \alpha, \beta, A, C) \\ \\ \left|y' - y - (-Cy + B - \beta x + \alpha y(y\cos\theta - x\sin\theta) + \alpha\cos\theta)\right| \leq T(G_{max}, L, \alpha, \beta, B, C) \end{cases}$$

avec

$$\begin{aligned} T(G_{max}, L, \alpha, \beta, A, C) = \; & G_{max}\Big[L^3\,\tfrac{\alpha^2}{2} + L^2\left(|C\alpha| + \tfrac{|\beta\alpha|}{2} + \tfrac{|\alpha|^3}{3}\right) \\ & + L\left(\tfrac{\alpha^2}{4}(6 + 3|\beta| + |C-1|) + |A\alpha| + \tfrac{|\beta C|}{2} + \tfrac{\beta^2}{4} + \tfrac{C^2}{2} + \tfrac{|\beta|^3}{12}\right) \\ & + |\alpha|\left(\tfrac{\beta^2}{2} + |\beta| + |C| + \tfrac{|\alpha A|}{2} + \tfrac{2\alpha^2}{3}\right) + |AC|\Big]. \end{aligned}$$

Démonstration. Considérons un mouvement de caméra $D = (R, t)$ décrit par les paramètres $(\theta, \alpha, \beta, A, B, C)$. La matrice de rotation s'écrit

$$\begin{aligned} R \; &= \begin{pmatrix} \cos\beta - (1-\cos\alpha)\sin\theta\sin(\theta-\beta) & -\sin\beta + (1-\cos\alpha)\sin\theta\cos(\theta-\beta) & \sin\theta\sin\alpha \\ \sin\beta + (1-\cos\alpha)\cos\theta\sin(\theta-\beta) & \cos\beta - (1-\cos\alpha)\cos\theta\cos(\theta-\beta) & -\cos\theta\sin\alpha \\ -\sin\alpha\sin(\theta-\beta) & \sin\alpha\cos(\theta-\beta) & \cos\alpha \end{pmatrix} \\ &= \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}. \end{aligned}$$

Les coefficients de R vérifient, par des développements limités au voisinage de 0 en α et β,

$$\begin{cases} a_1 = 1 + k_{a_1}, & k_{a_1} = o(\beta) + o(\alpha) & \text{et} & |k_{a_1}| \leq \beta^2/2 + \alpha^2/2(1+|\beta|) \\ a_2 = \beta + k_{a_2}, & k_{a_2} = o(\beta^2) + o(\alpha) & \text{et} & |k_{a_2}| \leq \beta^3/6 + \alpha^2/2(1+|\beta|) \\ a_3 = -\alpha\sin\theta + k_{a_3}, & k_{a_3} = o(\alpha^2) + o(\sqrt{|\alpha\beta|}) & \text{et} & |k_{a_3}| \leq \alpha^3/6 + |\alpha|(|\beta| + \beta^2/2) \\ b_1 = -\beta + k_{b_1}, & k_{b_1} = o(\beta^2) + o(\alpha) & \text{et} & |k_{b_1}| \leq \beta^3/6 + \alpha^2/2(1+|\beta|) \\ b_2 = 1 + k_{b_2}, & k_{b_2} = o(\beta) + o(\alpha) & \text{et} & |k_{b_2}| \leq \beta^2/2 + \alpha^2/2(1+|\beta|) \\ b_3 = \alpha\cos\theta + k_{b_3}, & k_{b_3} = o(\alpha^2) + o(\sqrt{|\alpha\beta|}) & \text{et} & |k_{b_3}| \leq \alpha^3/6 + |\alpha|(|\beta| + \beta^2/2) \\ c_1 = \alpha\sin\theta + k_{c_1} & k_{c_1} = o(\alpha^2) & \text{et} & |k_{c_1}| \leq |\alpha|^3/6 \\ c_2 = -\alpha\cos\theta + k_{c_2} & k_{c_2} = o(\alpha^2) & \text{et} & |k_{c_2}| \leq |\alpha|^3/6 \\ c_3 = 1 + k_{c_3} & k_{c_3} = o(\alpha) & \text{et} & |k_{c_3}| \leq |\alpha|^2/2. \end{cases}$$

En utilisant l'expression $(x', y') = \psi(x, y)$ donnée en (2.6), on a

$$\begin{cases} x' - x = \dfrac{x + \beta y - \alpha\sin\theta + A + o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})}{\alpha\sin\theta\, x - \alpha\cos\theta\, y + 1 + C + o(\alpha)} - x \\ \\ y' - y = \dfrac{y - \beta x + \alpha\cos\theta + B + o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})}{\alpha\sin\theta\, x - \alpha\cos\theta\, y + 1 + C + o(\alpha)} - y, \end{cases}$$

soit

$$\begin{cases} x'-x = & \Big(x+\beta y-\alpha\sin\theta+A+o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big) \\ & (1-C-\alpha\sin\theta\, x+\alpha\cos\theta\, y+o(\alpha)+o(C))-x \\ \\ y'-y = & \Big(y-\beta x+\alpha\cos\theta+B+o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big) \\ & (1-C-\alpha\sin\theta\, x+\alpha\cos\theta\, y+o(\alpha)+o(C))-y, \end{cases}$$

$$\begin{cases} x'-x = & -Cx+\beta y-\alpha\sin\theta+A-\alpha\sin\theta\, x^2+\alpha\cos\theta\, xy+o(\alpha)+o(\beta)+o(C) \\ & +o(\sqrt{|\alpha\beta|})+o(\sqrt{|C\beta|})+o(\sqrt{|C\alpha|})+o(\sqrt{|\alpha A|})+o(\sqrt{|CA|}) \\ \\ y'-y = & -Cy-\beta x+\alpha\cos\theta+B-\alpha\sin\theta\, xy+\alpha\cos\theta\, y^2+o(\alpha)+o(\beta)+o(C) \\ & +o(\sqrt{|\alpha\beta|})+o(\sqrt{|C\beta|})+o(\sqrt{|C\alpha|})+o(\sqrt{|\alpha B|})+o(\sqrt{|CB|}). \end{cases}$$

De plus, en utilisant les bornes de $|k_{a_1}|, |k_{a_2}|, \ldots, |k_{c_3}|$, on obtient

$$\begin{aligned} &\big|x'-x-\big(-Cx+\beta y-\alpha\sin\theta+A-\alpha\sin\theta\, x^2+\alpha\cos\theta\, xy\big)\,\big| \\ &= \left|\frac{-c_1x^2-c_2xy+(a_1-c_3-C)x+a_2y+a_3+A-(c_1x+c_2y+c_3+C)(A-Cx+\beta y+\alpha\cos\theta xy-\alpha\sin\theta x^2-\alpha\sin\theta)}{c_1x+c_2y+c_3+C}\right| \\ &\le G_{max} \quad \big|\, x^2(-c_1+Cc_1+\alpha\sin\theta c_3+\alpha\sin\theta C)-y^2\beta c_2+ \\ &\qquad xy(-c_2+Cc_2-\beta c_1-\alpha\cos\theta c_3-\alpha\cos\theta C)+x^2y(-c_1\alpha\cos\theta+c_2\alpha\sin\theta)+ \\ &\qquad x^3(\alpha\sin\theta c_1)-xy^2c_2\alpha\cos\theta+x(a_1-c_3-C-Ac_1+c_1\alpha\sin\theta+Cc_3+C^2)+ \\ &\qquad y(a_2-Ac_2+c_2\alpha\sin\theta-\beta c_3-\beta C)+a_3+A(1-c_3-C)+\alpha\sin\theta(c_3+C)\big| \end{aligned}$$

Comme $(x,y)\in[-L/2,L/2]^2$, on obtient

$$\begin{aligned} &\big|x'-x-\big(-Cx+\beta y-\alpha\sin\theta+A-\alpha\sin\theta\, x^2+\alpha\cos\theta\, xy\big)\,\big| \\ &\le G_{max} \quad \Big[L^3\,\tfrac{\alpha^2}{2}+L^2\left(|C\alpha|+\tfrac{|\beta\alpha|}{2}+\tfrac{|\alpha|^3}{3}\right) \\ &\qquad +L\left(\tfrac{\alpha^2}{4}(6+3|\beta|+|C-1|)+|A\alpha|+\tfrac{|\beta C|}{2}+\tfrac{\beta^2}{4}+\tfrac{C^2}{2}+\tfrac{|\beta|^3}{12}\right) \\ &\qquad +|\alpha|\left(\tfrac{\beta^2}{2}+|\beta|+|C|+\tfrac{|\alpha A|}{2}+\tfrac{2\alpha^2}{3}\right)+|AC|\Big]. \end{aligned}$$

De la même façon, on borne $\big|y'-y-\big(-Cy-\beta x+\alpha\cos\theta+B-\alpha\sin\theta\, xy+\alpha\cos\theta\, y^2\big)\,\big|$ en remplaçant A par B. □

Si les paramètres α, β, A, B, C et L sont suffisamment petits, le flot optique peut donc être approché par la somme de trois termes indépendants

$$\begin{cases} x'-x\simeq & -Cx+A & +\beta y & +\alpha x(y\cos\theta-x\sin\theta)-\alpha\sin\theta \\ y'-y\simeq & \underbrace{-Cy+B}_{(1)} & \underbrace{-\beta x}_{(2)} & \underbrace{+\alpha y(y\cos\theta-x\sin\theta)+\alpha\cos\theta}_{(3)}. \end{cases} \tag{2.11}$$

Le premier (1) est dû à la translation de la caméra, le deuxième (2) à la rotation R_β^k et le troisième (3) à la rotation $R_{\theta,\alpha}$. Ces trois termes sont des approximations des flots générés respectivement par la translation, la rotation R_β^k et la rotation $R_{\theta,\alpha}$. Par exemple, le flot généré par le mouvement de caméra $(\theta, \alpha, \beta, A, B, C) = (0, 0, 0, A, B, C)$, s'écrit, à partir de la formule (2.9)

$$\begin{cases} x' - x = \dfrac{x + A}{1 + C} - x = -Cx + A + o(C) \\ \\ y' - y = \dfrac{y + B}{1 + C} - y = -Cy + B + o(C). \end{cases}$$

De même, le flot généré par le mouvement de caméra $(\theta, \alpha, \beta, A, B, C) = (0, 0, \beta, 0, 0, 0)$ est égal à

$$\begin{cases} x' - x = x\cos\beta + y\sin\beta - x = \beta y + o(\beta) \\ y' - y = -x\sin\beta + y\cos\beta - y = -\beta x + o(\beta). \end{cases}$$

Enfin, le flot généré par le mouvement de caméra $(\theta, \alpha, \beta, A, B, C) = (\theta, \alpha, 0, 0, 0, 0)$ vaut

$$\begin{cases} x' - x = \dfrac{(\cos^2\theta + \sin^2\theta\cos\alpha)x + \cos\theta\sin\theta(1 - \cos\alpha)y - \sin\theta\sin\alpha}{x\sin\theta\sin\alpha - y\cos\theta\sin\alpha + \cos\alpha} - x \\ \\ y' - y = \dfrac{\cos\theta\sin\theta(1 - \cos\alpha)x + (\sin^2\theta + \cos^2\theta\cos\alpha)y + \cos\theta\sin\alpha}{x\sin\theta\sin\alpha - y\cos\theta\sin\alpha + \cos\alpha} - y, \end{cases}$$

soit

$$\begin{cases} x' - x = \dfrac{x - \alpha\sin\theta + o(\alpha)}{1 + x\alpha\sin\theta - y\alpha\cos\theta + 1 + o(\alpha)} - x \\ \\ y' - y = \dfrac{y + \alpha\cos\theta + o(\alpha)}{1 + x\alpha\sin\theta - y\alpha\cos\theta + 1 + o(\alpha)} - y \end{cases}$$

d'où

$$\begin{cases} x' - x = -\alpha\sin\theta - x^2\alpha\sin\theta + xy\alpha\cos\theta + o(\alpha) \\ \\ y' - y = \alpha\cos\theta - xy\alpha\sin\theta + y^2\alpha\cos\theta + o(\alpha). \end{cases}$$

Si les paramètres du mouvement et la taille de l'image sont suffisamment petits, le flot optique généré par un mouvement de caméra de paramètres $(\theta, \alpha, \beta, A, B, C)$ est donc approché par la somme des flots générés respectivement par la translation de paramètres (A, B, C), la rotation R_β^k et la rotation $R_{\theta,\alpha}$. L'approximation (2.11) du flot optique est à la base des algorithmes d'estimation du mouvement de la caméra présentés dans le chapitre suivant.

Remarques

– **Unités focale et pixellique**
Nous avons obtenu une approximation du flot optique pour $f_c = 1$. Si l'unité choisie pour le

repère de la caméra et pour celui de l'image n'est plus la longueur focale mais par exemple un pixel (une unité d'image) on a alors

$$\begin{cases} \dfrac{x'-x}{f_c} = & -C\dfrac{x}{f_c} + A + \beta\dfrac{y}{f_c} + \alpha\,\dfrac{x}{f_c}\,(-\dfrac{x}{f_c}\sin\theta + \dfrac{y}{f_c}\cos\theta) - \alpha\sin\theta + o(C) + o(\alpha) \\ & +o(\beta) + o(\sqrt{|\alpha A|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|AC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}) \\ \dfrac{y'-y}{f_c} = & -C\dfrac{y}{f_c} + B - \beta\dfrac{x}{f_c} + \alpha\,\dfrac{y}{f_c}\,(-\dfrac{x}{f_c}\sin\theta + \dfrac{y}{f_c}\cos\theta) + \alpha\cos\theta + o(C) + o(\alpha) \\ & +o(\beta) + o(\sqrt{|\alpha B|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|BC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}). \end{cases}$$

ce qui équivaut à

$$\begin{cases} x'-x = & -Cx + A\,f_c + \beta y + \alpha\,\dfrac{x}{f_c}\,(-x\sin\theta + y\cos\theta) - f_c\,\alpha\sin\theta + o(C) + o(\alpha) \\ & +o(\beta) + o(\sqrt{|\alpha A|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|AC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}) \\ y'-y = & -Cy + B\,f_c - \beta x + \alpha\,\dfrac{y}{f_c}\,(-x\sin\theta + y\cos\theta) + f_c\,\alpha\cos\theta + o(C) + o(\alpha) \\ & +o(\beta) + o(\sqrt{|\alpha B|}) + o(\sqrt{|\alpha C|}) + o(\sqrt{|BC|}) + o(\sqrt{|C\beta|}) + o(\sqrt{|\alpha\beta|}). \end{cases}$$

Les bornes $T(G_{max}, L, \alpha, \beta, A, C)$ et $T(G_{max}, L, \alpha, \beta, B, C)$ sont alors multipliées par f_c.

– **Valeurs des paramètres**

Comme nous l'avons déjà mentionné, considérer deux images successives d'une séquence vidéo suppose un mouvement de caméra entre les deux images très limité. Ceci implique des restrictions sur les valeurs possibles pour les six paramètres définis précédemment, exception faite pour l'angle θ, qui localise l'axe de la rotation $R_{\theta,\alpha}$ dans le plan (C, i, j). Le tableau (2.1) donne les intervalles de valeurs des paramètres, que nous avons obtenus expérimentalement en prenant des images variées, auxquelles nous avons appliqué diverses applications projectives définies par les six paramètres avec un angle de vue inférieur ou égal à 150° et une longueur focale égale à 1. Les valeurs retenues permettent de générer des séquences visuellement réalistes, c'est-à-dire en évitant les impressions de saccades visuelles trop importantes entre images. Mais suivant l'angle de vue choisi, les déformations produites par un ensemble de paramètres sont plus ou moins conséquentes ; les valeurs présentées dans le tableau génèrent des déformations acceptables pour une longueur focale unitaire et pour au moins un angle de vue inférieur à 150°.

Pour les bornes supérieures des valeurs des paramètres données dans le tableau, et avec $G_{max} = 4/3$, la constante de majoration T du théorème est égale à 10^{-4} $(6L^3 + 23L^2 + 81L + 32)$. Plus la taille des images est petite (c'est-à-dire l'angle de vue de la caméra

puisqu'on considère $f_c = 1$), plus la constante de majoration est faible. Par exemple, dans le cas d'un mouvement de translation pure, avec $A = B = 0.09$ et $C = 0.03$, l'approximation des composantes du flot optique (2.11) est d'ordre 10^{-2} voire 10^{-1} (car $A = 0.09$) et la majoration de chaque composante vaut $4.2\,10^{-3}$ pour $L = 1$ et $8.4\,10^{-3}$ pour $L = 8$. Dans le cas d'une rotation "purement" projective avec $\alpha = 0.01$, l'approximation de chaque composante du flot optique est d'ordre 10^{-2} et la majoration vaut $3\,10^{-4}$ pour $L = 1$ et $5.2\,10^{-3}$ pour $L = 4$. Pour $L = 8$, c'est-à-dire lorsque x et y sont grands, l'approximation de chaque composante est d'ordre 10^{-1} et la majoration vaut $3.6\,10^{-2}$.

Paramètre	Intervalle de valeurs
θ (radian)	$]-\pi, \pi]$
α (radian)	$[0, 3.10^{-2}]$
β (radian)	$[-5.10^{-2}, 5.10^{-2}]$
A,B	$[-9.10^{-2}, 9.10^{-2}]$
C	$[-3.10^{-2}, 3.10^{-2}]$

TABLEAU 2.1: *Valeurs prises par les six paramètres décrivant un mouvement de caméra.*

– **Action sur l'image des composantes de la déformation**
L'approximation obtenue est quadratique en x et y. Les termes en x^2, xy et y^2 sont dus à la rotation "purement" projective $R_{\theta,\alpha}$; ils sont d'autant plus grands que les valeurs de x et y sont importantes. Par exemple, considérons les valeurs des paramètres données dans le tableau (2.1) et L vérifiant $L \leq 8$ (correspondant à un angle de vue inférieur à 150°). Au centre de K, pour x et y d'ordre 10^{-1}, les termes en x^2, xy et y^2 sont négligeables devant les autres termes ; l'approximation du flot optique devient donc

$$\begin{pmatrix} -Cx + A + \beta y - \alpha \sin\theta \\ -Cy + B - \beta x + \alpha \cos\theta \end{pmatrix}.$$

Ceci signifie qu'au centre de l'image, la déformation est essentiellement affine. Plus précisément, c'est une similitude de paramètres de translation $(A - \alpha \sin\theta, B + \alpha \cos\theta)$, de rapport d'homothétie $\sqrt{C^2 + \beta^2}$ et d'angle de rotation $\arccos\left(-\frac{C}{\sqrt{C^2+\beta^2}}\right) \operatorname{sgn}(-\beta)$.
La figure (2.8) présente un exemple de flot optique observé en théorie entre deux images consécutives dans une séquence, c'est-à-dire que les valeurs des six paramètres (θ, α, β, A,

B, C) choisies appartiennent aux intervalles donnés dans le tableau (2.1). Elle illustre la remarque précédente. On a représenté sur cette figure deux composantes du flot optique, respectivement générées par la similitude et la rotation "purement" projective associées au mouvement de caméra, obtenues par la formule exacte (2.6) et par la formule approchée (2.11). On a choisi l'angle de vue de la caméra égal à 137°, ce qui équivaut à une image de taille 5.12×5.12 pour une longueur focale unitaire. On observe que la partie centrale de l'image du flot complet est très similaire à la partie centrale du flot généré par la similitude. De plus, sur le flot généré par la rotation "purement" projective, plus on s'éloigne du centre de l'image, plus la déformation est prononcée.

Le théorème suivant traite le cas général d'une scène filmée non plane.

Théorème 2.4 – *Soit $D = (R, t) \in SE(3)$, R paramétrée par (θ, α, β) avec $|\alpha| < 1$ et $|\beta| < 1$, $f_c = 1$ et K et K' les domaines de plus grande dimension L sur lesquels les images f et g sont définies. Soit Z définie sur K donnant les profondeurs des points projetés sur K, de bornes Z_{inf} et Z_{sup}. On suppose que D et Z vérifient les hypothèses (1) et (2). Si*

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} \leq 2\varepsilon$$

alors il existe $Z_0 > 0$ tel que $\forall (x, y) \in K$ et $(x', y') \in K'$ appariés par la formule (2.1),

$$\begin{cases} \left|x' - x - (-Cx + A + \beta y + \alpha x(y\cos\theta - x\sin\theta) - \alpha\sin\theta)\right| \leq T(G_{max}, L, \alpha, \beta, A, C) + \varepsilon \\ \\ \left|y' - y - (-Cy + B - \beta x + \alpha y(y\cos\theta - x\sin\theta) + \alpha\cos\theta)\right| \leq T(G_{max}, L, \alpha, \beta, B, C) + \varepsilon, \end{cases}$$

où $(A, B, C) = -\dfrac{1}{Z_0}\left(\langle t, R(i)\rangle, \langle t, R(j)\rangle, \langle t, R(k)\rangle\right)$.

Si le produit de la translation avec les variations des inverses des profondeurs est suffisamment faible, on peut substituer aux profondeurs $Z(x, y)$ une constante Z_0 dans les expressions (2.1), et ainsi les approcher par la fonction ψ associée au mouvement et à Z_0. D'après les valeurs des paramètres données dans le tableau (2.1), les composantes du flot optique sont chacune d'ordre 10^{-2} ; il est donc nécessaire d'avoir une erreur d'approximation lors de la substitution des profondeurs par une constante, au moins inférieure à 10^{-2}. Si le mouvement est suffisamment faible, on peut ensuite approcher le flot optique $\psi(x, y) - (x, y)$ par une expression quadratique en (x, y). Dans la pratique, le mouvement entre deux images consécutives étant très faible, si la caméra est suffisamment éloignée de la scène ou si les profondeurs de la scène varient peu, l'expression quadratique est une très bonne approximation du flot optique.

2.4.3 Relation entre l'approximation (2.11) du flot et la forme linéaire (1.9)

Dans le chapitre 1, nous avons mentionné une formule, donnée en (1.9), liant le flot optique à la vitesse de la caméra, formule fréquemment utilisée pour l'estimation du mouvement de la

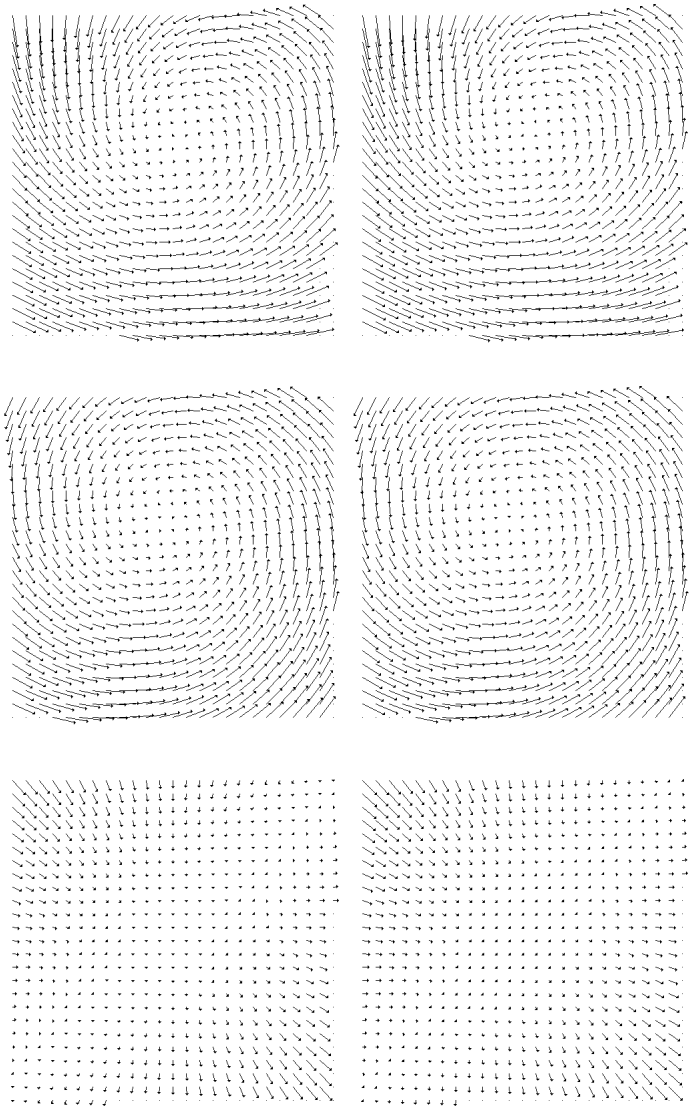

FIGURE 2.8: *À gauche, sont représentés les champs de flot optique générés par les formules exactes, et à droite par les formules d'approximation. Sur les trois premières lignes, de haut en bas, les flots sont générés par le mouvement de caméra de paramètres* $(\theta, \alpha, \beta, A, B, C) = (-\pi/4, 0.01, 0.04, 0.02, -0.01, 0.01)$*, la rotation plane d'angle* $\beta = 0.04$ *suivie de la translation de vecteur* $(0.02, -0.01, 0.01)$ *et la rotation* $R_{\theta,\alpha}$ *avec* $\theta = -\pi/4$ *et* $\alpha = 0.01$*. Les images ont été générées pour* $f_c = 1$ *et* $L = 5.12$ *et la norme des vecteurs de flot est multipliée par 5, afin de faciliter l'observation.*

caméra à partir du flot. Cette formule, linéaire en (v, ω), respectivement vitesses de déplacement et de rotation de la caméra, est aussi quadratique en (x, y). Comme nous avons obtenu précédemment en (2.11) une approximation du flot optique elle aussi quadratique en (x, y), on s'interroge naturellement sur les liens entre ces deux formules. Notons que l'on a généralisé le terme de flot optique au déplacement discret des points d'une image à l'autre.

La formule (1.9) exprime le flot optique $u(x, y, t)$ au point (x, y) en fonction de la profondeur $Z(x, y)$ du point projeté, de la vitesse de rotation $\omega(t) = (\omega_1(t), \omega_2(t), \omega_3(t))$ et de celle du déplacement $v(t) = (v_1(t), v_2(t), v_3(t))$ de la caméra
$u(x, y, t) =$

$$\begin{pmatrix} -\frac{v_1(t)}{Z(x,y)} - \omega_2(t) \\ -\frac{v_2(t)}{Z(x,y)} + \omega_1(t) \end{pmatrix} + \begin{pmatrix} \frac{v_3(t)}{Z(x,y)} & \omega_3(t) \\ -\omega_3(t) & \frac{v_3(t)}{Z(x,y)} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\omega_2(t) & \omega_1(t) & 0 \\ 0 & -\omega_2(t) & \omega_1(t) \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}.$$

La formule que nous avons obtenue en (2.11) exprime le déplacement $(x' - x, y' - y)$ du point (x, y) entre les deux images. Si on note dt l'intervalle de temps entre les deux acquisitions, alors

$$\begin{pmatrix} x' - x \\ y' - y \end{pmatrix} = u(x, y, t)\, dt.$$

Le flot $u(x, y, t)\, dt$ s'exprime donc en fonction des six paramètres du mouvement (θ, α, β, A, B, C) comme suit

$$u(x, y, t)\, dt = \begin{pmatrix} A - \alpha \sin\theta \\ B + \alpha \cos\theta \end{pmatrix} + \begin{pmatrix} -C & \beta \\ -\beta & -C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\alpha \sin\theta & \alpha \cos\theta & 0 \\ 0 & -\alpha \sin\theta & \alpha \cos\theta \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}.$$

Si les deux formules sont cohérentes, on doit avoir

$$\begin{pmatrix} \alpha \cos\theta \\ \alpha \sin\theta \\ \beta \end{pmatrix} = \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{pmatrix} dt,$$

et

$$\begin{pmatrix} A - \alpha \sin\theta \\ B + \alpha \cos\theta \\ C \end{pmatrix} = -\frac{1}{Z(x, y)} \begin{pmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{pmatrix} dt + \begin{pmatrix} -\omega_2(t) \\ \omega_1(t) \\ 0 \end{pmatrix} dt,$$

soit, comme on a considéré la profondeur de la scène constante et égale à Z_0 pour obtenir notre approximation,

$$\begin{pmatrix} \alpha \cos\theta \\ \alpha \sin\theta \\ \beta \end{pmatrix} = \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{pmatrix} dt \text{ et } \begin{pmatrix} A \\ B \\ C \end{pmatrix} = -\frac{1}{Z_0} \begin{pmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{pmatrix} dt.$$

Pour justifier cette correspondance, nous allons calculer les vitesses $\omega(t)$ et $v(t)$ associées à un mouvement de caméra paramétré par $(\theta, \alpha, \beta, A, B, C)$.

Théorème 2.5 – *Soit un mouvement de caméra $D \in SE(3)$. On suppose la scène plane et orthogonale à l'axe optique avant le déplacement, de profondeur Z_0 dans le repère de la caméra : on note $D = (\theta, \alpha, \beta, A, B, C)$. Si $\omega(t)$ et $v(t)$ sont les vitesses de rotation et translation de la caméra, et si dt est l'intervalle de temps entre les deux acquisitions consécutives, on a alors*

$$\omega(t)\ dt = \begin{pmatrix} \alpha\cos\theta \\ \alpha\sin\theta \\ \beta \end{pmatrix} + \left(o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})\right) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

et

$$\frac{v(t)}{Z_0}\ dt = -\begin{pmatrix} A \\ B \\ C \end{pmatrix} + \begin{pmatrix} o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|}) + o(\sqrt{|B\beta|}) + o(\sqrt{|C\alpha|}) \\ o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|}) + o(\sqrt{|A\beta|}) + o(\sqrt{|C\alpha|}) \\ o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|}) + o(\sqrt{|A\alpha|}) + o(\sqrt{|B\alpha|}) \end{pmatrix}.$$

Démonstration. 1) La vitesse de rotation ω associée à un mouvement de caméra ne dépend que de la rotation. Dans le chapitre 1, nous avons vu que l'on peut associer à une matrice du groupe $SO(3)$ un élément ω de l'algèbre de Lie correspondante $\mathfrak{so}(3)$ par

$$\omega = \frac{a}{\|u\|}\, u,$$

où u est un vecteur directeur de l'axe de rotation, non forcément unitaire, et a l'angle de la rotation. Cet élément de l'algèbre de Lie correspond au déplacement angulaire instantané entre les deux images, soit $\omega = \omega(t)\ dt$. Ici,

$$u = \begin{pmatrix} \sin\alpha\ (\cos\theta + \cos(\theta - \beta)) \\ \sin\alpha\ (\sin\theta + \sin(\theta - \beta)) \\ \sin\beta\ (1 + \cos\alpha) \end{pmatrix}$$

et, comme on sait que l'angle de rotation a est très faible (largement inférieur à $\pi/2$ en valeur absolue) car on considère deux images consécutives dans une séquence,

$$a = \arcsin\sqrt{\frac{1}{4}\sin^2\beta\ (1 + \cos\alpha)^2 + \frac{1}{2}\sin^2\alpha\ (1 + \cos\beta)}.$$

Comme

$$\|u\| = \sqrt{\sin^2\beta\ (1 + \cos\alpha)^2 + 2\sin^2\alpha\ (1 + \cos\beta)} = 2\sin a,$$

on a

$$\begin{aligned}\frac{a}{\|u\|} &= \frac{a}{2\sin a}\\ &= \frac{\arcsin\sqrt{\frac{1}{4}\sin^2\beta\,(1+\cos\alpha)^2+\frac{1}{2}\sin^2\alpha\,(1+\cos\beta)}}{2\sqrt{\frac{1}{4}\sin^2\beta\,(1+\cos\alpha)^2+\frac{1}{2}\sin^2\alpha\,(1+\cos\beta)}}\\ &= \frac{1}{2}+o(\sqrt{|\alpha|})+o(\sqrt{|\beta|}).\end{aligned}$$

Par des développements limités au voisinage de 0 en α et β, on obtient

$$u=\begin{pmatrix}2\alpha\cos\theta+o(\sqrt{|\alpha\beta|})+o(\beta)+o(\alpha^2)\\ 2\alpha\sin\theta+o(\sqrt{|\alpha\beta|})+o(\beta)+o(\alpha^2)\\ 2\beta+o(\alpha)+o(\beta^2)\end{pmatrix}$$

d'où

$$\frac{a}{\|u\|}\,u=\begin{pmatrix}\alpha\cos\theta\\ \alpha\sin\theta\\ \beta\end{pmatrix}+\Big(o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big)\begin{pmatrix}1\\1\\1\end{pmatrix}.$$

2) La vitesse de déplacement v associée à un mouvement de caméra dépend à la fois de la rotation et de la translation. La formule (1.7) du chapitre 1 donne une expression de la vitesse v associée à une vitesse de rotation ω et à une translation de caméra de vecteur t

$$v=\tau^{-1}(\omega)\,t=\left(I_3-\frac{1}{2}[\omega]_\times+\left(\frac{1}{\|\omega\|^2}-\frac{\sin\|\omega\|}{2\|\omega\|(1-\cos\|\omega\|)}\right)[\omega]_\times^2\right)\,t.$$

Ici encore, ω et v sont les déplacements angulaires et translationnels entre les deux images, et correspondent donc respectivement à $\omega(t)\,dt$ et $v(t)\,dt$.

Comme

$$\omega=\begin{pmatrix}\alpha\cos\theta\\ \alpha\sin\theta\\ \beta\end{pmatrix}+\Big(o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big)\begin{pmatrix}1\\1\\1\end{pmatrix},$$

on a

$$[\omega]_\times=\begin{pmatrix}0&-\beta&\alpha\sin\theta\\ \beta&0&-\alpha\cos\theta\\ -\alpha\sin\theta&\alpha\cos\theta&0\end{pmatrix}+\Big(o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big)\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix},$$

et

$$[\omega]_\times^2=\Big(o(\alpha)+o(\beta)+o(\sqrt{|\alpha\beta|})\Big)\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}.$$

Par un développement limité d'ordre 1 en $\|\omega\|$ au voisinage de 0, on a

$$\begin{aligned}\frac{1}{\|\omega\|^2} - \frac{\sin\|\omega\|}{2\|\omega\|(1-\cos\|\omega\|)} &= \frac{2(1-\cos\|\omega\|) - \|\omega\|\sin\|\omega\|}{2\|\omega\|^2(1-\cos\|\omega\|)} \\ &= \frac{2\left(\frac{\|\omega\|^2}{2} - \frac{\|\omega\|^4}{24}\right) - \|\omega\|\left(\|\omega\| - \frac{\|\omega\|^3}{6}\right) + o\left(\|\omega\|^5\right)}{\|\omega\|^4} \\ &= \frac{1}{12} + o\left(\|\omega\|\right).\end{aligned}$$

La matrice $\tau^{-1}(\omega)$ vérifie alors,

$$\tau^{-1}(\omega) = \begin{pmatrix} 1 & \frac{\beta}{2} & -\frac{\alpha\sin\theta}{2} \\ -\frac{\beta}{2} & 1 & \frac{\alpha\cos\theta}{2} \\ \frac{\alpha\sin\theta}{2} & -\frac{\alpha\cos\theta}{2} & 1 \end{pmatrix} + \left(o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})\right) \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

La vitesse v cherchée est égale à $\tau^{-1}(\omega)\ t$ et le vecteur de translation t est donné en fonction des six paramètres par (2.10)

$$t = Z_0\,\tilde{t} = Z_0(-AR(i) - BR(j) - CR(k)),$$

donc, en utilisant l'écriture de la matrice R en fonction de θ, α et β donnée en (2.8),

$$\frac{t}{Z_0} =$$

$$\left(-A\begin{pmatrix}1\\ \beta\\ -\alpha\sin\theta\end{pmatrix} - B\begin{pmatrix}-\beta\\ 1\\ \alpha\cos\theta\end{pmatrix} - C\begin{pmatrix}\alpha\sin\theta\\ -\alpha\cos\theta\\ 1\end{pmatrix}\right) + \left(o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})\right)\begin{pmatrix}1\\1\\1\end{pmatrix}.$$

Ainsi, v/Z_0 est égal à

$$\begin{aligned}\frac{v}{Z_0} &= \begin{pmatrix} -A + B\frac{\beta}{2} - C\frac{\alpha\sin\theta}{2} \\ -A\frac{\beta}{2} - B + C\frac{\alpha\cos\theta}{2} \\ A\frac{\alpha\sin\theta}{2} - B\frac{\alpha\cos\theta}{2} - C \end{pmatrix} + \left(o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})\right)\begin{pmatrix}1\\1\\1\end{pmatrix} \\ &= \begin{pmatrix}-A\\-B\\-C\end{pmatrix} + \begin{pmatrix} o(\sqrt{|B\beta|}) + o(\sqrt{|C\alpha|}) \\ o(\sqrt{|A\beta|}) + o(\sqrt{|C\alpha|}) \\ o(\sqrt{|C\alpha|}) + o(\sqrt{|B\alpha|}) \end{pmatrix} + \left(o(\alpha) + o(\beta) + o(\sqrt{|\alpha\beta|})\right)\begin{pmatrix}1\\1\\1\end{pmatrix}.\end{aligned}$$

□

L'approximation du flot optique obtenue en (2.11) est donc bien cohérente avec la formule (1.9). Dans le chapitre suivant, nous allons utiliser l'approximation (2.11) du flot optique entre deux images pour estimer les six paramètres $(\theta, \alpha, \beta, A, B, C)$ du mouvement de la caméra entre les deux acquisitions. Cette formule présente l'avantage d'être d'une part, quadratique en (x, y) et d'autre part, 1-linéaire en $(\alpha\cos\theta, \alpha\sin\theta, \beta, A, B, C)$.

2.5 Conclusion

Dans le contexte particulier de deux images f et g consécutives dans une séquence filmée par une caméra de longueur focale unitaire, et pour une scène et une translation de caméra t vérifiant

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} \leq 2\varepsilon$$

L étant la plus grande dimension des images, on peut approximer l'application liant l'image f à l'image g à ε près, en remplaçant les profondeurs par une même constante. Pour une faible valeur de ε (en pratique strictement inférieure à 10^{-2}) et si la scène est suffisamment éloignée de la caméra, ceci permet de modéliser les déformations d'images dans le groupe des recalages, isomorphe au groupe des déplacements de l'espace $SE(3)$. Ensuite, par une décomposition appropriée de la rotation de la caméra, nous écrivons la déformation entre deux images consécutives comme une déformation "purement" projective suivie d'une similitude. Cette décomposition nous permet d'approcher le flot optique par la somme de termes correspondants à la similitude d'une part et à la déformation "purement" projective d'autre part. Cette approximation du flot, quadratique en (x, y), est cohérente avec la formule mentionnée dans le chapitre 1, donnant le flot en fonction des vitesses v et ω de la caméra et elle aussi quadratique.

Dans le chapitre suivant, nous proposons d'estimer le mouvement de la caméra en utilisant l'approximation du flot optique obtenue ici.

Chapitre 3

Estimation d'un mouvement entre deux images consécutives

À partir de la décomposition d'un mouvement de caméra présentée dans le chapitre 2 et de l'approximation de la déformation sur l'image qui en découle (dans le contexte précisé dans le chapitre 2), on propose un algorithme d'estimation du mouvement de la caméra à partir de deux images consécutives dans une séquence.

La méthode présentée est directe et estime simultanément les six paramètres du mouvement. Elle utilise un algorithme d'estimation de mouvements paramétriques 2D, proposé par Odobez et Bouthémy dans [51] et implémenté dans le logiciel Motion2D. L'application de cet algorithme à l'estimation du mouvement de caméra a nécessité l'ajout d'un modèle dans le logiciel, modèle à six paramètres associé à la décomposition des déformations. Les performances de la méthode sont illustrées à travers les estimations de mouvement obtenues sur des séquences synthétiques et réelles, et quelques utilisations de ces estimations.

3.1 Estimation directe à partir des images

La méthode que nous proposons s'applique à l'estimation du mouvement d'une caméra entre deux images consécutives dans une séquence, notées f et g. Elle est basée sur l'approximation de la déformation entre f et g, obtenue dans le chapitre 2 pour une longueur focale égale à 1, une plus grande dimension L des domaines rectangulaires K et K' sur lesquels f et g sont définies, une scène de profondeurs bornées par Z_{inf} et Z_{sup} et une translation t de caméra vérifiant

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} < 2\varepsilon$$

pour une valeur de ε suffisamment faible ($\varepsilon < 10^{-2}$). Pour un mouvement de caméra paramétré par $(\theta, \alpha, \beta, A, B, C)$, un point (x, y) de l'image f est déplacé en (x', y') dans l'image g et le déplacement $(x' - x, y' - y)$ est approché par

$$\begin{pmatrix} A - \alpha \sin\theta \\ B + \alpha \cos\theta \end{pmatrix} + \begin{pmatrix} -C & \beta \\ -\beta & -C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\alpha \sin\theta & \alpha \cos\theta & 0 \\ 0 & -\alpha \sin\theta & \alpha \cos\theta \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}. \tag{3.1}$$

L'idée est d'estimer une déformation entre les images, afin d'en déduire le mouvement tridimensionnel de la caméra. Pour cela, nous allons recourir à la méthode d'estimation de mouvement paramétrique 2D d'Odobez et Bouthémy, décrite dans [51]. Ce choix a été fait après avoir testé et comparé différentes méthodes, exposées à la fin du chapitre.

3.1.1 Estimation de mouvements paramétriques 2D d'Odobez et Bouthémy

La méthode d'Odobez et Bouthémy permet de déterminer un mouvement 2D constant, affine ou quadratique entre deux images. C'est une méthode robuste, multirésolution, qui utilise seulement les gradients spatio-temporels de l'intensité. Elle a été implémentée dans un logiciel libre appelé Motion2D, disponible à l'adresse `http://www.irisa.fr/vista/Themes/Logiciel/Motion-2D/Motion-2D.html`.

Le principe est le suivant. Soient deux images f et g consécutives dans une séquence et $u(x, y)$ le flot optique entre les deux images au point (x, y) de f. Le déplacement $u(x, y)$ est supposé paramétrique et noté $u_\Theta(x, y)$ où Θ représente les paramètres du déplacement. Plusieurs modèles sont proposés ; nous allons présenter ici le plus général à 12 paramètres

$$u_\Theta(x, y) = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & q_3 \\ q_4 & q_5 & q_6 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}$$

où $\Theta = (c_1, c_2, a_1, \ldots, a_4, q_1, \ldots, q_6)$. Les auteurs considèrent la différence associée au modèle de mouvement paramétrique choisi, au point (x, y)

$$DF_{\Theta,\xi}(x, y) = g((x, y) + u_\Theta(x, y)) - f(x, y) + \xi$$

où u_Θ dépend des paramètres du mouvement à déterminer et ξ est un paramètre à estimer, correspondant à un éventuel changement global d'illumination entre les deux images.

La méthode classique des moindres carrés consiste à minimiser les carrés des différences sur (Θ, ξ)

$$\sum_{(x,y)\in S} DF_{\Theta,\xi}(x, y)^2$$

où S est le support choisi sur l'image. Mais ce procédé est instable si certaines observations sont aberrantes vis à vis du modèle choisi (ce qui se produit fréquemment dans les différences de niveaux de gris de pixels voisins). C'est pourquoi Odobez et Bouthémy préfèrent utiliser un M-estimateur, qui permet d'estimer le mouvement de façon robuste, tout en tolérant des données bruitées ou aberrantes. La fonctionnelle suivante est donc minimisée sur (Θ, ξ)

$$\sum_{(x,y)\in S} \rho(DF_{\Theta,\xi}(x, y), \Lambda).$$

Le M-estimateur ρ est une fonction paire, admettant un minimum unique en zéro et bornée par Λ (constante d'échelle choisie) pour les grandes valeurs de $DF_{\Theta,\xi}(x,y)$. L'utilisation de ρ pondère l'importance de l'observation en chaque pixel en fonction de sa conformité au modèle. Ceci explique le terme "M-estimateur" : la minimisation de la fonction correspond à l'estimation du maximum de vraisemblance si ρ est interprétée comme l'opposée de la log-vraisemblance associée au modèle. Une revue des différents M-estimateurs utilisés en vision est présentée dans la thèse de Black [6]. La fonction de Tukey est implémentée dans le logiciel Motion2D ; elle est définie par

$$\rho(t,\Lambda) = \begin{cases} \frac{t^2}{2}(\Lambda^4 - \Lambda^2 t^2 + \frac{t^4}{3}) & \text{si } |t| < \Lambda, \\ \\ \dfrac{\Lambda^6}{6} & \text{sinon} \end{cases}$$

et son graphe est présenté sur la figure (3.1).

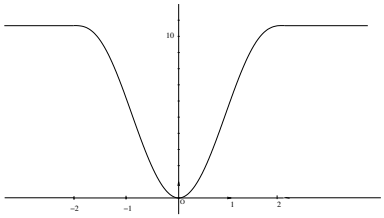

FIGURE 3.1: *Estimateur de Tukey (pour $\Lambda = 2$).*

La minimisation est réalisée par un schéma incrémental multirésolution. À chaque pas k, d'une résolution à une résolution plus fine, on a

$$\begin{cases} \widehat{\Theta}_{k+1} = \widehat{\Theta}_k + \Delta\Theta_k \\ \widehat{\xi}_{k+1} = \widehat{\xi}_k + \Delta\xi_k \end{cases}$$

où $\widehat{\Theta}_k$ et $\widehat{\xi}_k$ sont les valeurs estimées au pas k. Les valeurs $\Delta\Theta_k$ et $\Delta\xi_k$ sont déterminées comme suit ; par un développement limité à l'ordre 1 de $DF_{\Theta,\xi}(x,y)$ en $(\widehat{\Theta}_k, \widehat{\xi}_k)$, on obtient l'expression $r_{\Delta\Theta_k,\Delta\xi_k}(x,y)$

$$r_{\Delta\Theta_k,\Delta\xi_k}(x,y) = DF_{\widehat{\Theta}_k,\widehat{\xi}_k}(x,y) + \nabla g((x,y) + u_{\widehat{\Theta}_k}(x,y))\, u_{\Delta\Theta_k}(x,y) + \Delta\xi_k$$

où ∇g est le gradient spatial de la fonction d'intensité. À chaque pas k, on obtient une estimation de $\Delta\Theta_k$ et $\Delta\xi_k$ en minimisant la fonctionnelle

$$\sum_{(x,y)\in S} \rho\left(r_{\Delta\Theta_k,\Delta\xi_k}(x,y),\Lambda\right).$$

Cette erreur est minimisée en transformant le problème de M-estimation en un problème équivalent des moindres carrés pondérés et itérés [30]. Pour transformer le problème, on écrit

$$\sum_{(x,y)\in S} \rho\left(r_{\Delta\Theta_k,\Delta\xi_k}(x,y),\Lambda\right) = \sum_{(x,y)\in S} \frac{1}{2}\, w(x,y)\, r^2_{\Delta\Theta_k,\Delta\xi_k}(x,y).$$

Une condition nécessaire à la minimisation étant que la dérivée de l'erreur en chaque paramètre du mouvement et en ξ soit nulle, on obtient

$$\rho'(r_{\Delta\Theta_k(x,y),\Delta\xi_k},\Lambda) = w(x,y)\, r_{\Delta\Theta_k,\Delta\xi_k}(x,y)$$

$$\Rightarrow w(x,y) = \frac{\rho'(r_{\Delta\Theta_k,\Delta\xi_k}(x,y),\Lambda)}{r_{\Delta\Theta_k,\Delta\xi_k}(x,y)}.$$

Le problème est ainsi devenu un problème de moindres carrés pondérés et itérés, résolu à chaque pas k par l'algorithme de minimisations alternées suivant. Au départ, tous les poids sont initialisés à 1.

1. Pour $j \in \mathbb{N}$, à partir de l'erreur $r^j_{\Delta\Theta_k,\Delta\xi_k}$ commise au rang j, on calcule

$$w^j(x,y) = \frac{\rho'(r^j_{\Delta\Theta_k,\Delta\xi_k}(x,y),\Lambda)}{r^j_{\Delta\Theta_k,\Delta\xi_k}(x,y)}$$

2. on minimise par rapport à $\Delta\Theta_k, \Delta\xi_k$ par moindres carrés

$$\sum_{(x,y)\in S} w^j(x,y)\, \left(r^{j+1}_{\Delta\Theta_k,\Delta\xi_k}(x,y)\right)^2$$

3. jusqu'à atteindre la convergence.

La méthode des moindres carrés pondérés et itérés converge toujours, au moins vers un minimum local, comme l'a montré Osborne dans [53]. Odobez et Bouthémy montrent dans [51] que les résultats expérimentaux obtenus par cette méthode d'estimation multirésolution robuste (c'est-à-dire utilisant un estimateur robuste) sont meilleurs que ceux obtenus par une méthode multirésolution des moindres carrés.

3.1.2 Estimation du mouvement de la caméra

3.1.2.1 Ajout d'un modèle quadratique

L'implémentation de la méthode d'Odobez et Bouthémy dans le logiciel Motion2D permet d'estimer le mouvement bidimensionnel suivant un modèle choisi parmi trois modèles constants, dix affines et cinq quadratiques [65]. Notre approximation du flot optique en fonction du mouvement de la caméra suit un modèle quadratique à six paramètres, qui ne coïncide avec aucun

des cinq modèles quadratiques proposés. Nous avons donc ajouté au logiciel un modèle de flot adapté à l'estimation du mouvement entre deux images successives

$$u(x,y) = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}.$$

3.1.2.2 Conversion en mouvement de caméra

Cas d'une longueur focale unitaire Une fois les six paramètres (c_1, c_2, a_1, a_2, q_1, q_2) estimés, on en déduit facilement les paramètres du mouvement (θ, α, β, A, B, C) en identifiant l'expression de $u(x,y)$ ci-avant avec celle donnée en (3.1).

$$\begin{cases} \theta = \begin{cases} -\arctan(q_1/q_2) & \text{si } q_2 > 0 \\ -\arctan(q_1/q_2) + \pi & \text{si } q_2 < 0 \\ \pi/2 & \text{si } q_2 = 0 \text{ et } q_1 > 0 \\ -\pi/2 & \text{si } q_2 = 0 \text{ et } q_1 \leq 0 \end{cases} \\ \alpha = \sqrt{q_1^2 + q_2^2} \\ \beta = a_2 \\ A = c_1 + \alpha \sin\theta \\ B = c_2 - \alpha \cos\theta \\ C = -a_1. \end{cases}$$

Cas d'une longueur focale non unitaire Nous avons vu dans le chapitre 2 que la longueur focale agit sur l'échelle des déformations. En remplaçant x et y par x/f_c et y/f_c, on obtient l'approximation de la déformation entre les images dans le cas où f_c est différente de 1. L'approximation (3.1) obtenue pour une longueur focale unitaire est alors équivalente à $\begin{pmatrix} x'-x \\ \\ y'-y \end{pmatrix} \simeq$

$$\begin{pmatrix} f_c(A - \alpha\sin\theta) \\ \\ f_c(B + \alpha\cos\theta) \end{pmatrix} + \begin{pmatrix} -C & \beta \\ \\ -\beta & -C \end{pmatrix} \begin{pmatrix} x \\ \\ y \end{pmatrix} + \begin{pmatrix} -\frac{\alpha}{f_c}\sin\theta & \frac{\alpha}{f_c}\cos\theta & 0 \\ \\ 0 & -\frac{\alpha}{f_c}\sin\theta & \frac{\alpha}{f_c}\cos\theta \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}.$$

En identifiant avec l'expression du déplacement

$$u(x,y) = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix},$$

on obtient

$$\begin{cases} \theta = \begin{cases} -\arctan(q_1/q_2) & \text{si } q_2 > 0 \\ -\arctan(q_1/q_2) + \pi & \text{si } q_2 < 0 \\ \pi/2 & \text{si } q_2 = 0 \text{ et } q_1 > 0 \\ -\pi/2 & \text{si } q_2 = 0 \text{ et } q_1 \leq 0. \end{cases} \\ \alpha = f_c\sqrt{q_1^2 + q_2^2} \\ \beta = a_2 \\ A = c_1/f_c + \alpha \sin\theta \\ B = c_2/f_c - \alpha\cos\theta \\ C = -a_1 \end{cases}$$

où A, B et C sont donnés en unités de longueur focale.

Une fois les six paramètres $(\theta, \alpha, \beta, A, B, C)$ estimés, on doit pouvoir les convertir en un mouvement de caméra $D = (R, t)$. Comme on l'a vu dans le chapitre 2, la translation $\widetilde{t}$ et la rotation R de la caméra estimées sont obtenues en fonction des six paramètres par

$$\begin{cases} \widetilde{t} = -AR(i) - BR(j) - CR(k) \\ R = R_{\theta,\alpha}R_\beta^k \end{cases}$$

où $\widetilde{t}$ est la translation divisée par la profondeur moyenne Z_0 de la scène filmée.

3.2 Validité du modèle

3.2.1 Précision des estimations obtenues

Afin d'évaluer quantitativement les performances de notre méthode, nous allons étudier sa précision et sa robustesse sur des images consécutives de films de scènes 2D. Lors de la construction de tels films (décrite plus en détail ci-après), la profondeur des points est considérée constante ; ces données correspondent donc exactement au cadre dans lequel nous avons proposé la résolution du problème (car $\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}} = 0$).

Nous avons créé trois films à partir d'une image réelle, présentée sur la figure (3.2) de taille 384×288 et de mouvements de caméra, c'est-à-dire d'ensembles de six paramètres (θ, α, β, A, B, C), générés aléatoirement mais dont les valeurs respectent les ordres de grandeur donnés dans le tableau (2.1) du chapitre 2. Nous avons choisi un angle de vue égal à 90°, soit une longueur focale égale à 192 pixels. Pour chaque film, nous avons appliqué la déformation projective associée au premier mouvement de caméra à l'image réelle en utilisant une interpolation bilinéaire ; puis, nous avons composé chaque nouveau mouvement avec l'ancien afin de toujours appliquer la déformation à l'image initiale. La première séquence est simulée à partir de mouvements quelconques, la deuxième uniquement par des rotations et la troisième seulement par des translations. Les images sont déformées en utilisant les formules exactes. Les résultats des

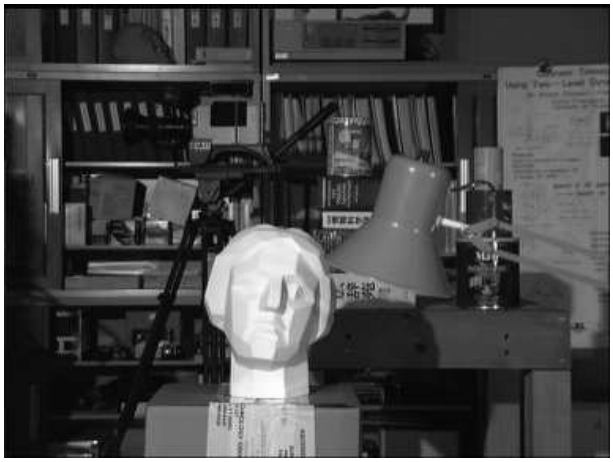

FIGURE 3.2: *Image utilisée pour créer les films tests.*

estimations des mouvements sont donnés dans le tableau (3.1). Afin d'éviter l'apparition des bords de l'image initiale, nous avons extrait dans chaque séquence le centre de l'image de taille 284×188.

	Erreur direction translation	Erreur direction axe de rotation	Erreur angle de rotation	
			absolue	relative
Mouvements quelconques	9.7°	17.3°	0.03°	2.2%
Translations pures	4.5°	-	0.01°	-
Rotations pures	-	18.2°	0.002°	0.1%

TABLEAU 3.1: *Résultats des estimations du mouvement de la caméra sur trois séquences synthétiques comportant chacune 200 images ; la première est simulée par des mouvements quelconques, la deuxième et la troisième sont simulées respectivement par des translations et par des rotations de caméra. Les erreurs données dans le tableau sont les erreurs moyennes calculées sur chaque séquence.*

Quel que soit le mouvement de la caméra, les estimations des directions des translations sont correctes à quelques degrés près (en moyenne 10° près pour les mouvements quelconques) et celle des angles de rotation à quelques centièmes de degrés (sachant que l'amplitude des rotations est de quelques degrés). L'estimation des axes de rotation est cependant moins précise : à une ou

deux dizaines de degrés près en général. Ces erreurs s'expliquent par la difficulté à séparer les composantes rotationnelle et translationnelle des déformations. Une modification de la direction de l'axe optique peut produire un effet sur l'image très proche de celui généré par une translation. Par exemple, une rotation autour de l'axe j (c'est-à-dire l'axe (CY)) engendre une déformation très voisine de celle produite par une translation de direction i (axe (CX)). La figure (3.3) illustre cette ambiguité. En conséquence, dans le cas de rotations pures, l'estimation des paramètres de

FIGURE 3.3: *Images de Lena obtenues, à gauche après une rotation d'axe j, avec $\theta = \pi/2$ et $\alpha = 0.022$, à droite après une translation d'axe i, avec $A = 0.03$. On a ici supposé l'angle de vue égal à 60°. Les déformations observées sont très voisines, ce qui rend difficile l'estimation.*

la rotation est corrompue par l'estimation d'une translation non nulle. Et inversement, dans le cas de translations pures, l'estimation de la direction de la translation est altérée par l'estimation d'une rotation non nulle. Néanmoins, les résultats d'estimation du mouvement que nous avons obtenus séparent les composantes du mouvement, non pas parfaitement mais de façon satisfaisante, comme le montre notamment l'estimation de l'angle de rotation. En particulier, dans le cas de mouvements uniquement translationnels ou rotationnels, les paramètres sont encore mieux estimés que dans le cas de mouvements composés.

3.2.2 Robustesse au bruit

On cherche ici à évaluer la robustesse de la méthode au bruit. Pour cela, on ajoute aux 200 images de la séquence générée précédemment par des mouvements quelconques de caméra, des quantités variables de bruit, impulsionnel ou gaussien.

L'ajout de bruit impulsionnel sur une image f de la séquence consiste à transformer le niveau de gris $f(x, y)$ d'un pixel (x, y) de la façon suivante

$$f(x,y) \longmapsto \begin{cases} f(x,y) & \text{si } n(x,y) = 0 \\ b(x,y) & \text{si } n(x,y) = 1 \end{cases}$$

où n est un champ de variables aléatoires indépendantes de Bernoulli de paramètre p et b est un champ de variables aléatoires, indépendantes entre elles et des variables du champ n, uniformément distribuées sur l'intervalle $[\min f, \max f]$. On affecte au paramètre p des valeurs de 0 à 0.3 ; le bruit touche ainsi de 0 à 30% des pixels de chaque image.

L'ajout de bruit gaussien correspond à la transformation suivante

$$f(x, y) \longmapsto f(x, y) + g(x, y),$$

où g est un champ de variables aléatoires gaussiennes, indépendantes et identiquement distribuées, centrées et d'écart-type σ. Ici, nous avons fait varier le paramètre σ de 0 à 50.

La figure (3.4) présente la première image du film généré précédemment, à laquelle on a appliqué un bruit gaussien d'écart-type 50 et un bruit impulsionnel touchant 30% des pixels. Les résultats des erreurs moyennes sur la séquence bruitée en fonction de la nature du bruit

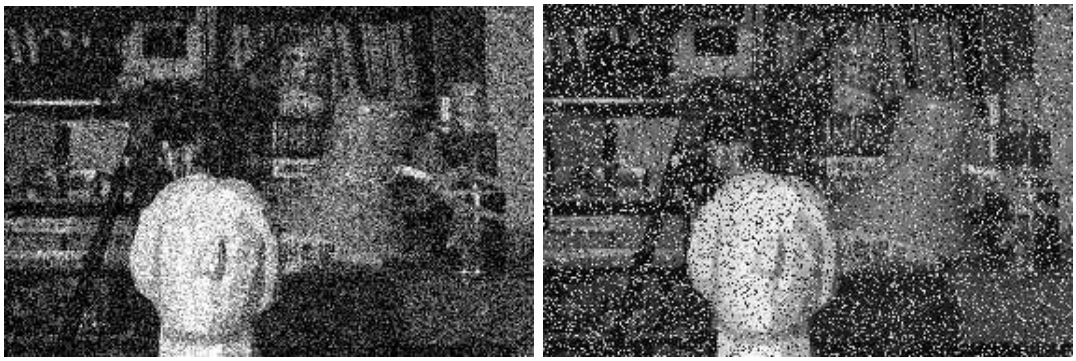

FIGURE 3.4: *Illustration de l'ajout de bruit sur l'image initiale utilisée pour créer les films tests. À gauche, un bruit gaussien d'écart-type* 50 *est appliqué, à droite un bruit impulsionnel touchant* 30% *des pixels.*

et de son amplitude sont présentés sur la figure (3.5). Pour les deux types de bruit, les erreurs d'estimation augmentent peu : elles restent très voisines des erreurs observées sans bruit, moins de 15 degrés pour la direction de la translation, au plus quelques dixièmes de degrés (pour l'ajout de bruit impulsionnel) pour l'angle de rotation. La méthode est donc robuste, grâce à l'utilisation du M-estimateur ; elle fournit encore de bons résultats même lorsque la quantité de bruit impulsionnel ajouté est importante.

3.2.3 Résultats sur des séquences 3D

Nous présentons ici des résultats d'estimation du mouvement de la caméra, obtenus sur des séquences 3D réelles. Pour la plupart des séquences utilisées, nous ne connaissons pas la longueur focale associée à la caméra ayant filmé la scène. Suivant les séquences, nous supposerons l'angle de vue compris entre 60 et 90°. Ces choix ne modifient pas ou peu les résultats présentés ; ils seront cependant discutés pour chaque séquence.

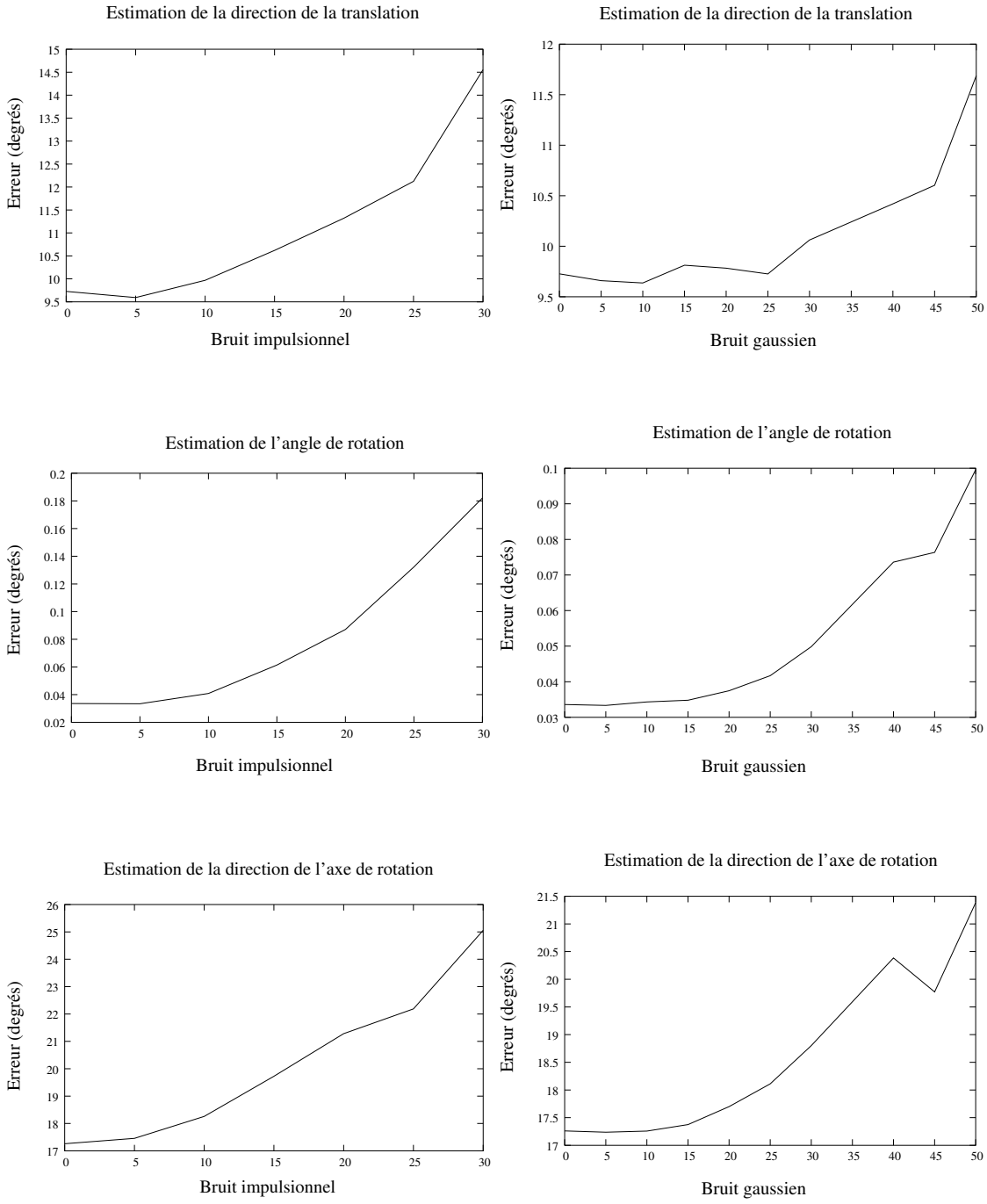

FIGURE 3.5: *Erreurs moyennes d'estimation du mouvement de la caméra sur la séquence de* 200 *images bruitées. On a ajouté à la séquence initiale, à gauche, un bruit impulsionnel touchant de* 0 *à* 30% *des pixels et à droite, un bruit gaussien d'écart-type allant de* 0 *à* 50.

3.2.3.1 Séquence "Soda-can"

Dans cette séquence, disponible à l'adresse `www.cs.brown.edu/~black/images.html`, la caméra, en mouvement de translation horizontale, filme une cannette de soda. Deux des 10 images de la séquence sont présentées sur la figure (3.6). Dans le cas d'une translation non nulle, notre méthode d'estimation s'applique si les variations de l'inverse de la profondeur sont suffisamment faibles. Ici, nous ne possédons pas d'information sur la structure de la scène mais il apparaît que celle-ci est constituée de deux plans distincts, la cannette et le fond texturé, reliés par une table. Il n'est pas certain que la variation de profondeur entre le premier et le dernier plan soit suffisamment faible pour que la séquence corresponde au cadre défini dans le chapitre précédent, cependant, sans obtenir des résultats parfaits, l'estimation du mouvement est correcte.

FIGURE 3.6: *Images 1 et 5 de la séquence "Soda-can"; la caméra effectue une translation horizontale.*

Les résultats obtenus par notre méthode d'estimation sont présentés dans le tableau (3.2). Nous avons supposé l'angle de vue égal à 60°, soit, comme les images sont de taille 201 × 201, une longueur focale égale à 174 pixels. Entre chaque couple d'images, la direction de la translation est correctement estimée à quelques degrés près (de 4 à 7°) ; l'angle de la rotation estimé (qui devrait être nul) est de l'ordre de l'erreur calculée précédemment sur les séquences synthétiques. L'estimation du mouvement est stable : les erreurs sont à peu près constantes sur la séquence. Avec une longueur focale différente, les résultats sont légèrement modifiés : les erreurs sur l'estimation de la direction de la translation et l'angle de rotation augmentent respectivement de 2° et 0.01° pour un angle de vue de 45° (soit $f_c = 242.6$ pixels) et diminuent respectivement de 3° et 0.01° pour un angle de vue de 90° (soit $f_c = 100.5$ pixels).

3.2.3.2 Test d'incrustation

Nous appliquons ici notre méthode d'estimation du mouvement à une séquence réelle filmée dans un bureau. Ici encore, nous ne possédons pas d'information sur les profondeurs exactes

Mouvement	Erreur d'estimation dans la direction de la translation	Angle de rotation
1	4.1°	0.07°
2	5.4°	0.05°
3	4.9°	0.07°
4	4.5°	0.06°
5	5.7°	0.06°
6	5.9°	0.06°
7	7.0°	0.08°
8	5.9°	0.07°
9	6.3°	0.06°

TABLEAU 3.2: *Erreur d'estimation de la direction de la translation et de l'angle de rotation estimé sur la séquence "Soda-can".*

de la scène 3D et il n'est pas évident que tout couple d'images consécutives satisfasse au cadre défini dans le chapitre 2. En particulier, le plafond et les murs latéraux (figure (3.7)) génèrent des différences de profondeur importantes. Mais l'éloignement de la scène autorise de plus grandes variations de profondeurs et rapproche du cadre défini au début du chapitre.

Nous avons appliqué notre méthode d'estimation du mouvement à la séquence en supposant l'angle de vue égal à 90° (ce qui permet de calculer la longueur focale en unités de pixels). Comme nous ne disposons pas de la donnée du mouvement de la caméra ayant généré ce film, nous allons illustrer la qualité de l'estimation du mouvement par un test d'incrustation d'une forme géométrique. Nous avons inséré dans une image de la séquence (sur une zone de l'image à peu près plane et orthogonale à l'axe optique) un rectangle noir, déformé ensuite par les applications projectives associées aux estimations du mouvement. Le rectangle initial est déformé par le mouvement correspondant à la composition des mouvements estimés entre l'image initiale et l'image dans laquelle il est inséré.

Des images extraites de la nouvelle séquence, contenant l'incrustation du rectangle grâce à l'estimation du mouvement de la caméra, sont présentées sur la figure (3.7). Remarquons qu'il est possible d'effectuer directement l'augmentation sans se préoccuper de la profondeur des objets de la scène car les variations relatives de profondeur, dans le repères associé à la caméra, au niveau de la zone d'insertion, sont faibles. On observe sur les résultats obtenus que l'orientation du rectangle ajouté, parallèle lors de son insertion dans la première image à l'arête entre le plafond et le mur, suit l'orientation de cette arête tout au long du film, ce qui illustre l'évaluation correcte des rotations de la caméra. De plus, la position du rectangle, sans être parfaitement précise, reste vraisemblable pendant toute la séquence.

Avec la même technique, on a masqué le tableau d'affichage du bureau par une affiche de cinéma. En déformant l'affiche avec les mouvements estimés et en la superposant au film original

FIGURE 3.7: *En haut : insertion d'un rectangle sur la première image du film. Au-dessous, les images* 10*,* 20*,* 30*,* 40*,* 55 *et* 70 *de la nouvelle séquence en utilisant les estimations de mouvement de la caméra.*

on obtient une séquence augmentée, dont quelques images sont présentées sur la figure (3.8). La déformation de l'affiche est réalisée par interpolation bilinéaire.

Remarque – Le but de cette expérience est l'illustration de la bonne estimation du mouvement de la caméra et non la réalisation d'une véritable augmentation de la séquence, avec un objet tridimensionnel. La réalité augmentée est en effet un champ de recherche à part entière, dont le but est la modification d'un film de façon que l'objet ajouté s'intègre naturellement à la séquence, comme s'il avait été présent lors du tournage (voir par exemple [1]).

FIGURE 3.8: *Disparition du tableau d'affichage au profit d'une affiche de cinéma. En haut : l'insertion de l'affiche sur la première image. Au-dessous, les images* 10*,* 20*,* 30*,* 40 *et* 45 *de la nouvelle séquence obtenue en déformant l'affiche avec les estimations des mouvements de la caméra.*

3.3 Temps de calcul

La méthode mise au point présente l'avantage important d'être très rapide. Le temps de calcul dépend du nombre d'images et des dimensions de celles-ci. Le tableau (3.3) précise les temps de calculs de la méthode sur les séquences proposées précédemment. Le processeur utilisé est un Pentium M à 1.8 GHz.

Séquence	Nombre d'images	Dimensions	Temps de calcul
séquence synthétique	130	284×188	7.70 s
séquence du bureau	100	176×144	4.88 s
séquence "Soda-can"	11	201×201	1.16 s

TABLEAU 3.3: *Temps de calcul de la méthode d'estimation du mouvement de caméra sur les séquences précédemment utilisées.*

3.4 Application : construction de mosaïques

Comme on suppose que deux images consécutives dans une séquence sont liées par une transformation plane (associée à un mouvement de caméra), il est possible de recaler les images les unes sur les autres. On utilise alors la composition des déformations dans le groupe des recalages. Nous illustrons notre propos avec le film du bureau présenté précédemment.

Le mosaïquage consiste à choisir deux images dans une séquence, par exemple I_n et I_p, avec $n < p$. Avec l'estimation du mouvement de la caméra sur toute la séquence, on peut calculer le mouvement entre les instants n et p. En effet, la méthode fournit les $p-n$ applications projectives entre les images n et p ; comme elles sont associées à un mouvement de caméra, on peut composer les applications dans le groupe des recalages, et obtenir le mouvement de la caméra entre I_n et I_p et par là l'application projective liant les deux images. En appliquant cette transformation à I_n, on obtient une image contenant une partie commune avec I_p et une partie agrandissant le champ, qui est juxtaposée à I_p. L'image résultante correspond à l'image observée au temps p, mais avec un champ de vision plus large.

Pour illustrer la qualité de l'estimation du mouvement de caméra par notre méthode, on utilise les paramètres estimés pour construire plusieurs mosaïquages. Les figures (3.9) et (3.10) présentent des panoramas réalisés en recalant, sur la première figure, deux images sur une troisième et sur la deuxième figure, quatre images sur une cinquième.

Notons que pour une scène 3D, le mosaïquage n'est théoriquement possible que si le point de vue n'est pas modifié, c'est-à-dire si la caméra n'effectue que des rotations et aucune translation, à cause des variations de profondeurs et des effets de parallaxe. Entre des images consécutives dans la séquence, on pouvait supposer ces effets négligeables, car le produit de la norme de la translation (très limitée) avec l'amplitude de l'inverse des profondeurs était faible. Cependant, entre des images éloignées dans la séquence, une telle hypothèse n'est plus possible car la translation est plus importante. De ce fait, sur les panoramas obtenus, on observe des recalages d'images bien ajustés à certaines profondeurs de la scène, et décalés à d'autres. Néanmoins, la scène filmée étant suffisamment plane, nous obtenons des vues panoramiques intéressantes, offrant une plus grande perspective de la scène.

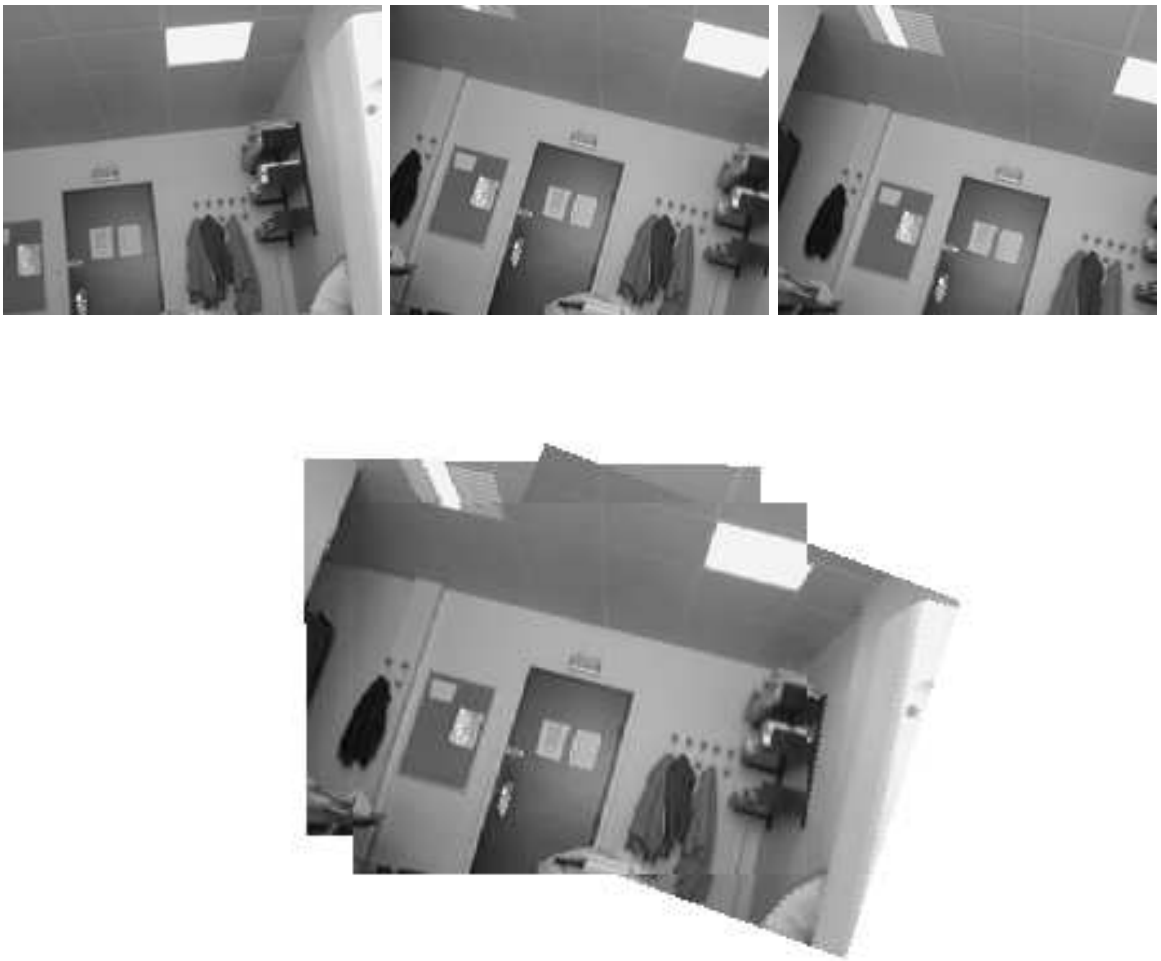

FIGURE 3.9: *En haut, les images* 20, 35 *et* 50 *de la séquence du bureau et au-dessous, la vue reconstruite du point de vue de l'image* 35.

FIGURE 3.10: *En haut, les images* 10, 30, 60, 70 *et* 80 *de la séquence. En bas, le panorama construit du point de vue de l'image* 60.

3.5 Limites du cadre d'application

3.5.1 Profondeur de la scène

Nous illustrons ici l'influence des variations relatives des profondeurs de la scène 3D filmée sur les résultats obtenus par notre méthode d'estimation de mouvement. Pour une scène tridimensionnelle donnée, plus la caméra est éloignée et moins les variations de profondeurs altèrent l'estimation car on peut alors plus facilement approximer la scène par un plan orthogonal à l'axe optique.

L'influence de la distance de la caméra à la scène est illustrée par l'estimation du mouvement d'une caméra sur des séquences synthétiques d'images de scènes 3D issues de l'ensemble SOFA (Sequences for Optical Flow Analysis). C'est un ensemble de séquences conçues par le groupe de vision par ordinateur de l'université d'Heriot-Watt pour tester les applications en analyse de mouvement. Chaque séquence est fournie avec les paramètres intrinsèques et extrinsèques et le mouvement de la caméra. Nous utilisons les séquences 5 et 6 de cet ensemble ; les mouvements de la caméra sont très simples : une translation de direction k pour SOFA5 et une rotation d'axe k suivie d'une translation de direction k pour SOFA6. La scène filmée est constituée de quatre cylindres empilés placés sur un fond texturé. Les figures (3.11) et (3.12) montrent les deux premières et les deux dernières images de chaque séquence. Le tableau (3.4) précise les

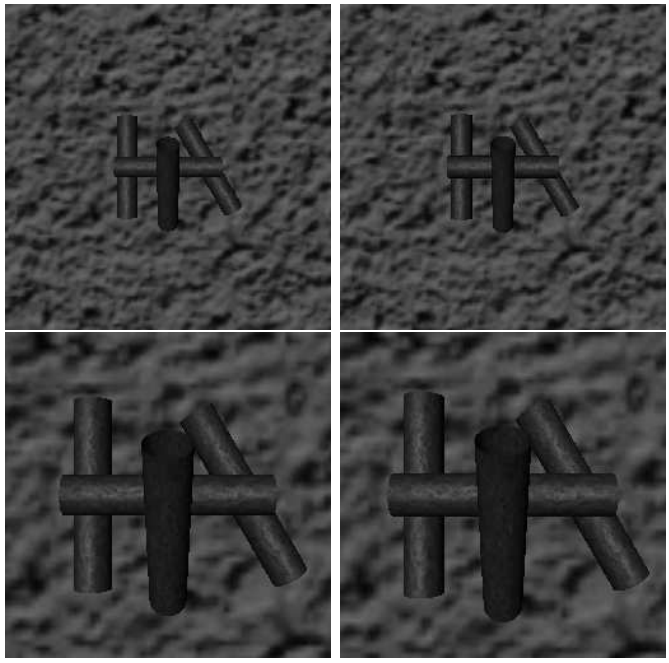

FIGURE 3.11: *En haut, les deux premières images de la séquence SOFA5 et en bas, les deux dernières.*

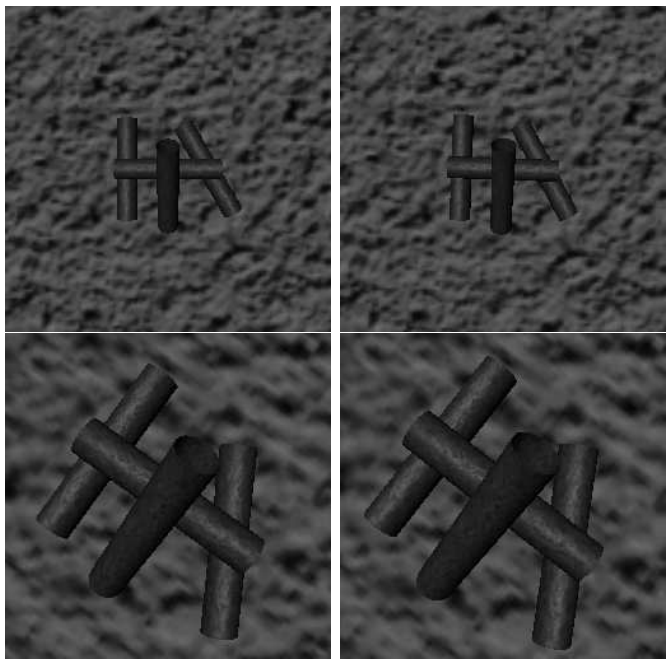

FIGURE 3.12: *En haut, les deux premières images de la séquence SOFA6 et en bas, les deux dernières.*

différences $1/Z_{inf} - 1/Z_{sup}$ en unités de longueur focale au cours des séquences SOFA5 et SOFA6, ainsi que la quantité $\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \frac{(L+1)\, G_{max}}{2}$. Plus cette quantité est faible et plus on se rapproche du cadre d'application de la méthode. Comme la caméra s'approche de la scène, les différences augmentent au cours du temps. Notons qu'on a $L = 0.83\, f_c$, ce qui équivaut à un angle de vue égal à 45°.

	$\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}$	$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \frac{(L+1)\, G_{max}}{2}$
Image 1	0.0062	0.0076
Image 10	0.0112	0.0137
Image 20	0.0293	0.0357

TABLEAU 3.4: *Variations relatives de l'inverse des profondeurs dans les séquences SOFA5 et SOFA6. Les profondeurs Z_{inf} et Z_{sup} sont exprimées en unités de longueur focale dans le repère associé à la caméra lors de l'acquisition des images. Les quantités $\|t\|$ et L sont également exprimées en unités de longueur focale.*

Les tableaux (3.5) et (3.6) fournissent les erreurs d'estimation du mouvement entre des images consécutives, au début, au milieu et à la fin de chaque séquence. La méthode utilisée pour estimer le mouvement est la même que celle utilisée précédemment : on ne suppose aucun type de mouvement a priori. Tout au long de la séquence SOFA5, la direction de la translation

	Erreur direction de la translation	Erreur angle de rotation
Entre les images 1 et 2	0.12°	0.0005°
Entre les images 10 et 11	0.17°	0.0018°
Entre les images 19 et 20	0.55°	0.019°
Erreurs moyennes	0.42°	0.014°

TABLEAU 3.5: *Erreurs d'estimation sur la séquence SOFA5, comportant* 20 *images. Le mouvement est constant sur la séquence : c'est une translation de direction k (la caméra s'approche de la scène).*

	Erreur direction de la translation	Erreur direction axe de rotation	Erreur angle de rotation	
			absolue	relative
Entre les images 1 et 2	0.23°	0.001°	0.051°	2.5%
Entre les images 10 et 11	0.38°	0.491°	0.068°	3.4%
Entre les images 19 et 20	0.97°	1.08°	0.094°	4.7%
Erreurs moyennes	0.39°	0.269°	0.069°	3.4%

TABLEAU 3.6: *Erreurs d'estimation sur la séquence SOFA6, comportant* 20 *images. Le mouvement est constant sur la séquence : une translation de direction k (la caméra s'approche de la scène) et une rotation d'axe k et d'angle* 2°.

est très bien estimée, beaucoup mieux que sur les séquences synthétiques du paragraphe 3.2.1. Ceci est imputable à la simplicité du mouvement et en particulier à la fixité de l'axe optique. On

observe cependant que lorsqu'on se rapproche de la scène, l'erreur d'estimation de la translation augmente légèrement, et avec elle l'angle de la rotation estimée (qui devrait être nul). Sur la séquence SOFA6, la direction de la translation est toujours très bien estimée ; en revanche, les erreurs sur l'estimation de la direction de l'axe de rotation et de l'angle de la rotation augmentent significativement à mesure que la caméra se rapproche des cylindres.

Bien que les erreurs augmentent lorsqu'on se rapproche de la scène (car on s'éloigne du contexte dans lequel nous avons travaillé), la méthode mise au point permet de conclure pour des mouvements simples (notamment lorsque l'axe optique est fixe) même dans des cas sortant du cadre d'application.

3.5.2 Objet en mouvement dans la scène

La méthode d'estimation proposée s'applique au cas d'une caméra en mouvement dans un environnement statique. Néanmoins, si un objet a un mouvement propre dans la scène, et si sa taille est limitée relativement à celle de l'image, le mouvement de la caméra peut être correctement estimé. Ceci est rendu possible par la globalité de la méthode : les informations en tous les points des images sont prises en compte lors de la minimisation via le M-estimateur.

La robustesse de la méthode au mouvement propre d'un objet est illustrée dans le tableau (3.7). Nous avons estimé le mouvement de la caméra sur la séquence "Street" disponible à l'adresse `www.cs.otago.ac.nz/research/vision/Research/OpticalFlow/opticalflow.html`. Dans cette séquence, la caméra a un mouvement de rotation autour de l'axe i (soit (CX)) et suit une voiture en mouvement. Deux images issues du film sont présentées sur la figure (3.13). On sup-

FIGURE 3.13: *Images 5 et 15 issues de la séquence "Street". La séquence comporte 20 images de dimensions* 200×200. *Les mouvements observés sur les images ont deux sources : le mouvement propre de la voiture et le mouvement de la caméra (qui effectue une rotation d'axe i).*

pose l'angle de vue égal à 60°. Le tableau (3.7) présente l'erreur d'estimation relative à l'axe de la rotation ainsi que l'angle de rotation entre chaque couple d'images. L'angle de rotation est d'ordre 0.2, valeur significativement supérieure à l'erreur moyenne calculée sur la séquence

synthétique dans le tableau (3.1), ce qui permet de considérer la rotation comme significative. L'axe de la rotation estimé est l'axe i à quelques degrés près (de 1 à 4). L'estimation de l'axe de rotation est donc correcte et stable ; notre méthode s'applique malgré l'objet en mouvement. Ceci est rendu possible par la taille limitée de la voiture qui occupe une surface maximale d'environ 5% de chaque image.

Avec des longueurs focales différentes, les résultats sont très proches ; l'erreur sur la direction de l'axe de rotation diminue de 0.3° en moyenne pour un angle de vue égal à 90° et augmente de 0.1° pour un angle de vue de 45°. Quant à l'angle de rotation estimé, il est d'ordre 0.1° et 0.3°, respectivement pour un angle de vue égal à 90° et 45°. Ceci n'est pas surprenant car l'angle de rotation autour de i vaut $\alpha = f_c\sqrt{q_1^2 + q_2^2}$: la valeur de l'angle estimé augmente donc avec la valeur de la longueur focale.

3.6 Autres méthodes

L'objectif de l'étude proposée dans ce chapitre était d'utiliser l'approximation quadratique de la déformation ψ pour estimer un mouvement de caméra, sur des séquences synthétiques et réelles. Nous avons étudié et testé différentes méthodes basées sur l'approximation de la déformation entre deux images consécutives avant de choisir la méthode que nous avons présentée ci-avant. Ici, nous en décrivons deux que, pour des raisons explicitées ci-après, nous n'avons pas retenues.

3.6.1 Régression multilinéaire sur le flot optique

L'idée est ici d'utiliser l'approximation quadratique des déformations, non plus directement sur le contenu des images f et g mais sur le flot optique entre ces images. Cette méthode nécessite donc le calcul du flot optique, effectué par la méthode de Weickert et Schnörr [74].

Une fois que l'on dispose du flot optique, il est assez naturel de vouloir obtenir les estimateurs des moindres carrés des paramètres à partir de l'approximation du flot, linéaire en $(\beta, A, B, C, \alpha\cos\theta, \alpha\sin\theta)$. Soit $u(x, y)$ le vecteur de flot calculé au point (x, y). En posant $\mu = \alpha\cos\theta$ et $\nu = \alpha\sin\theta$, on a l'approximation suivante,

$$u(x,y) = \begin{pmatrix} u_1(x,y) \\ u_2(x,y) \end{pmatrix} \simeq \begin{pmatrix} -Cx + A + \beta y + \mu xy - \nu(x^2+1) \\ -Cy + B - \beta x + \mu(y^2+1) - \nu xy \end{pmatrix}.$$

Afin d'utiliser toute l'information disponible, on minimise l'erreur quadratique

$$\begin{aligned} E(\beta, A, B, C, \mu, \nu) = \sum_{(x,y)\in f} \Big[&\big(u_1(x,y) - (-Cx + A + \beta y + \mu xy - \nu(x^2+1))\big)^2 \\ &+ \big(u_2(x,y) - (-Cy + B - \beta x + \mu(y^2+1) - \nu xy)\big)^2 \Big]. \end{aligned}$$

Pour cela, on effectue une régression multilinéaire sur les paramètres β, A, B, C, μ et ν. En dérivant E en les six paramètres et en mettant les dérivées à 0, on obtient un système de six

Mouvement	Erreur direction axe de rotation	Angle de rotation
1	2.9°	0.20°
2	3.1°	0.20°
3	2.7°	0.21°
4	0.8°	0.20°
5	2.8°	0.21°
6	2.7°	0.21°
7	0.9°	0.21°
8	2.9°	0.20°
9	2.9°	0.20°
10	3.6°	0.20°
11	2.9°	0.20°
12	0.6°	0.21°
13	3.2°	0.20°
14	2.7°	0.21°
15	2.9°	0.20°
16	1.4°	0.19°
17	3.2°	0.18°
18	3.1°	0.21°
19	2.8°	0.21°
Moyenne	2.5°	0.20°

TABLEAU 3.7: *Erreur d'estimation sur la direction de l'axe de rotation et angle de rotation estimé sur la séquence "Street" de* 20 *images.*

équations à six inconnues. La résolution du système conduit à des formules fermées pour les estimateurs des six paramètres.

Le calcul du flot optique est assez coûteux en temps de calcul. Mais une fois le flot optique estimé, cette méthode a l'avantage d'être très rapide. Elle donne de très bons résultats sur des flots optiques synthétiques. Comme toute méthode de régression, elle est robuste à l'ajout de bruit gaussien sur les composantes du flot et plus sensible à l'ajout de bruit impulsionnel. Malheureusement, sur des séquences réelles, l'estimation du flot optique sur l'image entière présente trop de valeurs aberrantes pour que la méthode fournisse de bons résultats. Sur des séquences réelles et avec un flot optique estimé, la méthode utilisant directement le contenu des images donne de bien meilleurs résultats.

3.6.2 Estimation de la similitude et raffinement

Nous avons présenté une variante de cette méthode dans [37]. Rappelons l'approximation du flot optique exprimée en pixels sur l'image, pour un mouvement de caméra petit, un angle de vue inférieur à 150° et une amplitude des variations des inverses des profondeurs de la scène limitée $\begin{pmatrix} x'-x \\ y'-y \end{pmatrix} \simeq$

$$f_c \begin{pmatrix} A - \alpha \sin\theta \\ B + \alpha \cos\theta \end{pmatrix} + \begin{pmatrix} -C & \beta \\ -\beta & -C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{f_c} \begin{pmatrix} -\alpha \sin\theta & \alpha\cos\theta & 0 \\ 0 & -\alpha\sin\theta & \alpha\cos\theta \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}.$$

Compte-tenu des valeurs prises par les six paramètres (chapitre 2, tableau 2.1), le centre de l'image f, correspondant à des valeurs de x et y d'ordre 1 ou 10, est principalement déformé par la composante affine de l'approximation tandis que la composante quadratique n'intervient que lorsque l'un des termes xy, x^2 ou y^2 est au moins d'ordre 10^3. Cette remarque est à la base de la méthode présentée ici. L'estimation est réalisée en deux temps : les quatre paramètres de la déformation affine d'abord (qui est en fait une similitude) puis les deux paramètres θ et α de la rotation "purement" projective.

La première estimation est effectuée à partir des centres des images f et g, principalement déformés par la similitude de paramètres de translation $f_c(A - \alpha\sin\theta)$, $f_c(B + \alpha\cos\theta)$, de rapport d'homothétie $\sqrt{\beta^2 + C^2}$ et d'angle de rotation $\arccos\left(-\frac{C}{\sqrt{C^2+\beta^2}}\right) \operatorname{sgn}(-\beta)$. Nous avons expérimenté deux techniques d'estimation de similitude. L'algorithme de Pascal Monasse, décrit dans [12, 48] consiste à apparier les ensembles de niveau des images extraites puis à faire voter chaque appariement pour une similitude. Les ensembles de niveau (plus exactement, les composantes connexes des ensembles de niveau dont les "trous" ont été remplis) sont appariés à partir de vecteurs caractéristiques dont les composantes sont invariantes par similitude. Ces composantes sont construites en pratique à partir des moments d'ordre 0 à 3 des ensembles de niveau. La similitude estimée est celle ayant reçu le plus grand nombre de votes. La deuxième technique expérimentée est la méthode d'Odobez et Bouthémy avec le modèle à 4 paramètres adapté à l'estimation d'homothétie translation et de rotation plane. Les résultats des deux méthodes sont comparables mais le coût de calcul de l'appariement des ensembles de niveau est nettement plus élevé que celui du logiciel Motion2D ; c'est pourquoi on choisit cette deuxième solution.

La deuxième étape consiste à estimer les paramètres θ et α de la rotation $R_{\theta,\alpha}$. On estime ces deux angles à partir du flot optique V mesuré entre les images f et g. On réécrit l'approximation du flot V,

$$V(x,y) \simeq V_s(x,y) + V_p(x,y)$$

où V_s est le flot généré par la similitude et V_p correspond à la composante quadratique de l'approximation. À ce stade de l'algorithme, on connaît une estimation des paramètres de la similitude $\hat{s}$ et on peut donc calculer le flot optique $V_{\hat{s}}$ généré par celle-ci. On en déduit que

la différence des flots $V - V_{\hat{s}}$ est due à la composante quadratique V_p. On note cette différence $V - V_{\hat{s}} = V_{\hat{p}} = \left(V^1_{\hat{p}}, V^2_{\hat{p}}\right)$. Comme $V_p(x, y)$ est linéaire en $(\alpha \sin\theta/f_c, \alpha \cos\theta/f_c)$, on peut estimer par régression linéaire les paramètres $\mu = \alpha \cos\theta/f_c$ et $\nu = \alpha \sin\theta/f_c$ en minimisant

$$\sum_{(x,y)\in f} \Big[\big(V^1_{\hat{p}}(x,y) - (-\nu x^2 + \mu xy)\big)^2 + \big(V^2_{\hat{p}}(x,y) - (-\nu xy + \mu y^2)\big)^2 \Big].$$

Cette étape peut être interprétée comme le raffinement de l'estimation, permettant d'ajuster la déformation aux bords de l'image. En effet, la déformation au centre de l'image est déjà globalement estimée par la similitude. La détermination de α et θ ajuste le "recalage" en périphérie de l'image f sur l'image g donnée. À partir des estimations de μ, ν, des paramètres de la similitude et de la connaissance de la longueur focale, on déduit le mouvement de la caméra estimé.

Cette méthode est une hybride des deux autres méthodes présentées, utilisant la technique d'estimation d'Odobez et Bouthémy sur le contenu des images pour estimer la similitude d'une part, mettant en oeuvre une régression linéaire sur le flot optique d'autre part. Les résultats obtenus par cette méthode sont nettement moins bons que ceux fournis par la méthode retenue (tableau 3.8), du fait du biais introduit par la succession d'estimations des paramètres et de l'utilisation du flot optique qui, s'il est calculé sur la séquence, présente un bruit important. De plus, l'étape de calcul du flot optique, par la méthode de Weickert et Schnörr par exemple, allonge considérablement le temps de calcul global de la méthode.

Dans la littérature, Irani, Rousso et Peleg ont une approche comparable dans [36]. Ils estiment un mouvement 2D entre deux images et utilisent le flot résiduel pour en déduire une partie du mouvement de la caméra. Plus précisément, ils détectent d'abord une région plane de l'image puis, avec le flot optique calculé, estiment le mouvement paramétrique 2D de la région, suivant le modèle quadratique associé à la forme bilinéaire (1.9). Un nouveau flot est alors calculé en soustrayant au flot optique initial le flot généré par le mouvement 2D. La composante rotationnelle est absente de ce flot résiduel car la rotation agit sur les points des images indépendamment des profondeurs des points 3D projetés. Les vecteurs du flot résiduel sont donc dirigés vers ou envers le foyer d'expansion, ce qui permet de déduire la direction de la translation. La rotation est ensuite calculée à partir des paramètres du mouvement 2D estimé de la région et de la translation.

3.6.3 Comparaison avec la méthode retenue

Le tableau (3.8) présente les estimations des mouvements de caméra sur le film déjà utilisé dans la partie 3.2.1. Nous avons créé ce film en déformant une image à partir de mouvements aléatoires, c'est-à-dire d'ensembles de six paramètres vérifiant les ordres de grandeur donnés dans le tableau (2.1). L'angle de vue est égal à 90°. Les erreurs moyennes d'estimation du mouvement sur les 200 images par les trois méthodes sont données.

La méthode de régression linéaire sur le flot optique théorique fournit les meilleurs résultats ; cependant, on dispose rarement de cette donnée (on en dispose si l'on connaît déjà le mouvement

	Erreur direction de la translation	Erreur direction axe de rotation	Erreur angle de rotation	
			absolue	relative
Régression linéaire sur le flot optique théorique	8.2°	20°	0.001°	0.09%
Régression linéaire sur le flot optique calculé par W. et S.	56.4°	59.5°	1.47°	96.7%
Estimation de similitude et raffinement	14.5°	27.1°	0.07°	4.6%
Méthode retenue	9.7°	17.3°	0.03°	2.2%

TABLEAU 3.8: *Résultats des estimations du mouvement de la caméra sur la séquence synthétique comportant 200 images par la méthode de régression linéaire sur le flot optique théorique, puis sur le flot estimé par la méthode de Weickert et Schnörr, par la méthode estimant une similitude puis les paramètres de la rotation "purement" projective et enfin par la méthode retenue (estimation directe à partir des images par la méthode d'Odobez et Bouthémy). Les erreurs données dans le tableau sont les erreurs moyennes calculées sur les 199 estimations.*

de la caméra !). Les résultats obtenus par cette même méthode sur le flot optique calculé par la méthode de Weickert et Schnörr sur la séquence, sont beaucoup trop imprécis pour être utilisables. En particulier, l'erreur sur l'angle de rotation est très importante ; ceci est dû au calcul des vecteurs de flot, de norme très inférieure à celle du déplacement réel. On obtient de meilleurs résultats avec la méthode estimant la similitude puis les paramètres θ et α ; ils sont cependant moins précis que ceux donnés par la méthode d'Odobez et Bouthémy appliquée directement sur les images et estimant les paramètres de l'approximation quadratique des déplacements des points.

Nous avons donc retenu cette dernière méthode pour l'estimation du mouvement d'une caméra, car non seulement elle donne de meilleurs résultats que les deux autres mais elle est aussi plus rapide. En effet, c'est une méthode directe qui ne nécessite ni le calcul du flot optique ni la mise en correspondance des points, ce qui limite le coût global de calcul ; à cet avantage s'ajoute l'implémentation par un schéma multirésolution qui augmente la vitesse d'exécution. De plus, nous avons vu qu'elle fournissait de bons résultats sur les données bruitées car elle utilise des estimateurs robustes. Enfin, les résultats obtenus sont bons, aussi bien sur des séquences synthétiques que sur des séquences réelles.

3.7 Conclusion

Dans ce chapitre, nous avons proposé une méthode d'estimation d'un mouvement de caméra entre deux images consécutives d'une séquence, dans le cas où, t étant la translation de la caméra, Z la profondeur de la scène et L la plus grande dimension des images, exprimées en unités de longueur focale,

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} < 2\varepsilon$$

pour $\varepsilon < 10^{-2}$. La détermination du mouvement est basée sur le développement quadratique en (x, y) de la déformation entre deux images obtenu dans le chapitre précédent. En composant les déformations estimées dans le groupe des recalages, on accède au mouvement entre des images éloignées dans la séquence, ce qui nous a permis de présenter des résultats de mosaïquage dans ce chapitre. Au prix d'un cadre d'application en théorie assez restrictif, la méthode proposée est très rapide et robuste, grâce à l'utilisation du logiciel Motion2D, adapté à notre modèle à six paramètres. Pour que la méthode s'applique, il faut théoriquement que la scène filmée soit plane et orthogonale à l'axe optique mais en pratique, si la condition donnée sur Z, L et t donnée ci-avant est vérifiée, le mouvement est bien estimé. La borne supérieure 10^{-2} peut même être légèrement dépassée car comme la méthode est globale, les différences de profondeurs se compensent.

Cette méthode présente les avantages inhérents aux méthodes directes : elle ne nécessite ni calcul de flot optique ni appariement de points préalablement à son application et elle utilise l'information en tous les points des deux images considérées ; grâce à cette globalité, la méthode est robuste à la présence d'un objet en mouvement propre dans la scène (de taille limitée relativement aux dimensions des images). Comparée à d'autres méthodes directes, elle estime simultanément et rapidement tous les paramètres du mouvement de la caméra.

Chapitre 4

Estimation itérative des profondeurs et du mouvement de la caméra

Nous utilisons l'estimation du mouvement d'une caméra présentée dans le chapitre précédent pour obtenir une carte des profondeurs de la scène filmée. Pour cela, nous appliquons un algorithme probabiliste, appelé Belief Propagation, déjà utilisé en stéréovision sur des images rectifiées. Ici, la Belief Propagation est mise en oeuvre directement, sans rectification, à partir de deux images consécutives d'une séquence sur laquelle on a estimé le mouvement de la caméra. Ce mouvement estimé sert d'initialisation à la carte des profondeurs de la scène, fournissant en chaque pixel une distribution de probabilité sur les profondeurs relatives du point projeté en ce pixel. À partir des profondeurs estimées, on détermine le mouvement de la caméra sur des zones de profondeurs voisines, ce qui conduit à une estimation plus précise. Ce procédé itératif permet d'étendre le cadre d'estimation défini dans le chapitre précédent, c'est-à-dire d'estimer le mouvement même si les variations de profondeurs sont importantes.

4.1 Présentation de l'algorithme de Belief Propagation

La méthode appelée Belief Propagation par Pearl dans [55], est une méthode générale permettant de résoudre des problèmes d'inférence dans des domaines aussi variés que la physique statistique, l'intelligence artificielle, le décodage des codes correcteurs d'erreurs, la vision par ordinateur... Elle a été découverte séparément dans les différents domaines sous les noms d'algorithme de Viterbi, algorithme de décodage itératif pour les codes de Gallager et les turbocodes, approche par matrice de transfert en physique... et algorithme de Belief Propagation de Pearl pour les réseaux bayésiens. Cette technique de résolution s'applique dès que le problème peut être représenté par un modèle graphique, c'est-à-dire un ensemble de noeuds ayant des relations entre eux ; l'état le plus probable de ces noeuds est alors recherché.

4.1.1 Modélisation markovienne d'une image

L'image est constituée d'un ensemble fini S de sites s, correspondant aux pixels. À chaque site, est associé un descripteur, qui peut être un niveau de gris, un label dans le cas d'une image segmentée ou une donnée plus complexe. L'idée sous-jacente de la modélisation probabiliste de l'image est que le descripteur d'un site n'est pas significatif en lui-même mais dans ses interactions avec les descripteurs des sites voisins. C'est pourquoi on associe à un ensemble de sites un système de voisinage.

Définition 4.1 – *Un système de voisinage* Γ *sur un ensemble de sites* S *est une collection de sous-ensembles* $(\Gamma_s)_{s\in S}$ *de* S *vérifiant*

- $s \notin \Gamma_s$
- $t \in \Gamma_s \Rightarrow s \in \Gamma_t$.

Les systèmes de voisinage les plus souvent utilisés sont la 4-connexité et la 8-connexité : un site a alors 4 ou 8 voisins, comme illustré sur les figures (4.1) et (4.2). À partir d'un système de voisinage, on déduit un système de cliques. Une clique est soit un singleton de S, soit un ensemble de sites tous voisins entre eux.

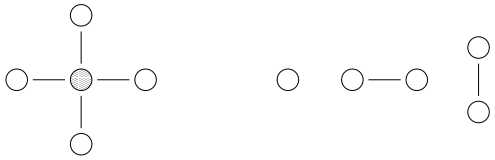

FIGURE 4.1: *Le système de voisinage induit par la* 4*-connexité et les trois types de cliques associées.*

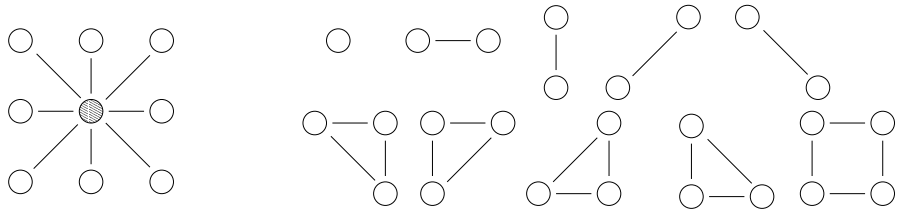

FIGURE 4.2: *Le système de voisinage induit par la* 8*-connexité et les dix types de cliques associées.*

Les interactions locales entre sites voisins s'expriment sous la forme de potentiels de clique. L'énergie globale U de l'image est la somme des énergies de l'ensemble $\mathcal{C}$ des cliques de l'image

$$U = \sum_{c\in\mathcal{C}} U_c$$

et l'énergie en un site s est la somme des énergies des cliques auquelles le site appartient

$$U_s = \sum_{c \in \mathcal{C} \text{ tq } s \in c} U_c.$$

Pour définir les champs de Markov, on considère une image $\mathtt{x}$ comme la réalisation d'un champ aléatoire $\mathtt{X}$. On note $\mathtt{X} = \{\mathtt{X}_s, s \in S\}$ le champ aléatoire, $\mathtt{x} = \{\mathtt{x}_s, s \in S\}$ une réalisation et E l'ensemble des valeurs prises par les variables aléatoires $\mathtt{X}_s$. Ainsi, $P(\mathtt{X}_s = \mathtt{x}_s)$ est la probabilité que la valeur du descripteur au site s soit $\mathtt{x}_s$ et $P(\mathtt{X} = \mathtt{x})$ est la probabilité d'observer l'image $\mathtt{x}$. Le lien entre le descripteur en un site s et le reste de l'image est mesuré par la probabilité conditionnelle

$$P(\mathtt{X}_s = \mathtt{x}_s | \mathtt{X}^s = \mathtt{x}^s) \text{ où } \mathtt{X}^s = \{\mathtt{X}_t,\ t \in S, t \neq s\}.$$

Définition 4.2 – *Soit un système de voisinage* Γ. *Un champ aléatoire* $\mathtt{X}$ *de taille* N *est un champ de Markov pour le voisinage* Γ *si et seulement si*

$$\begin{cases} \forall \mathtt{x}_s \in E,\ P(\mathtt{X}_s = \mathtt{x}_s | \mathtt{X}^s = \mathtt{x}^s) = P(\mathtt{X}_s = \mathtt{x}_s | \mathtt{X}_t = \mathtt{x}_t, t \in \Gamma_s) \\ \\ \forall \mathtt{x} \in E^N,\ P(\mathtt{X} = \mathtt{x}) > 0. \end{cases}$$

Ainsi, un champ aléatoire $\mathtt{X}$ est un champ de Markov pour un voisinage Γ si aucune configuration n'est interdite et si la probabilité conditionnelle locale en un site n'est fonction que de la configuration du voisinage du site considéré. Sur des domaines bornés, tous les champs aléatoires sont évidemment markoviens pour un voisinage suffisamment grand. Mais l'intérêt de cette modélisation tient justement aux champs de Markov associés à des voisinages restreints, permettant de réaliser des calculs rapides ; la plupart des images naturelles ou texturées vérifient l'hypothèse markovienne pour des voisinages tels que la 4 ou la 8-connexité.

Les champs de Markov associés au système de voisinage de la 4-connexité sont appelés champs de Markov d'ordre 1. Ils fournissent des modèles théoriques particulièrement intéressants pour certains problèmes de vision par ordinateur [20]. Dans ces problèmes, on veut le plus souvent inférer une réalisation cachée d'un champ de Markov.

4.1.2 Problème d'inférence dans un cadre bayésien

Les problèmes d'inférence sont posés en ces termes : on dispose d'une observation $\mathtt{y}$ sur un ensemble de sites S. On note $\mathtt{y} = \{\mathtt{y}_s, s \in S\}$ l'image observée et on souhaite déterminer une image inconnue $\mathtt{x} = \{\mathtt{x}_s, s \in S\}$. L'image $\mathtt{x}$ est considérée comme la réalisation d'un champ aléatoire $\mathtt{X}$ et l'image $\mathtt{y}$ celle d'un champ aléatoire $\mathtt{Y}$. Le champ $\mathtt{Y}$ dépend de $\mathtt{X}$ à travers une probabilité conditionnelle connue $P(\mathtt{Y} = \mathtt{y} | \mathtt{X} = \mathtt{x})$.

La restauration bayésienne de l'image $\mathtt{x}$ est basée sur la probabilité *a posteriori*

$$P(\mathtt{X} = \mathtt{x} | \mathtt{Y} = \mathtt{y}) \propto P(\mathtt{X} = \mathtt{x}) P(\mathtt{Y} = \mathtt{y} | \mathtt{X} = \mathtt{x}).$$

L'estimateur $\hat{\mathtt{x}}$ maximisant cette probabilité est appelé estimateur du maximum *a posteriori* (MAP). Deux hypothèses sont classiquement ajoutées ; d'une part, le champ $\mathtt{X}$ est supposé markovien pour un système de voisinage Γ (le plus souvent la 4-connexité pour les images), d'autre part, sachant $\mathtt{X} = \mathtt{x}$, les variables aléatoires $\mathtt{Y}_s$ sont supposées indépendantes et de loi conditionnelle ne dépendant que de la réalisation $\mathtt{x}_s$, c'est-à-dire

$$\begin{aligned} P(\mathtt{Y} = \mathtt{y} | \mathtt{X} = \mathtt{x}) &= \prod_{s \in S} P(\mathtt{Y}_s = \mathtt{y}_s | \mathtt{X}_s = \mathtt{x}_s) \\ &= \prod_{s \in S} \phi(\mathtt{x}_s, \mathtt{y}_s) \end{aligned}$$

La figure (4.3) illustre le problème à résoudre compte-tenu de ces hypothèses, dans le cas d'un champ $\mathtt{X}$ markovien pour la 4-connexité.

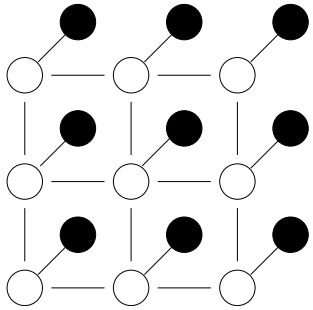

FIGURE 4.3: *Représentation du problème d'inférence dans les champs de Markov d'ordre 1. Les ronds noirs sont les informations* $\mathtt{y}_s$ *observées et les ronds blancs les valeurs* $\mathtt{x}_s$ *à estimer. Les traits représentent les interactions entre les sites.*

4.1.3 Description de la Belief Propagation

L'algorithme de Belief Propagation nécessite la connaissance de la fonction ϕ et la définition de la compatibilité entre les sites voisins du champ de Markov $\mathtt{X}$, c'est-à-dire les énergies des cliques d'ordre 2 pour la 4-connexité. La connaissance *a priori* sur le champ $\mathtt{X}$ s'écrit

$$P(\mathtt{X} = \mathtt{x}) = \prod_{s \in S,\, t \in \Gamma_s} \psi_{st}(\mathtt{x}_s, \mathtt{x}_t).$$

La fonction ψ_{st} est une matrice carrée de taille $|E| \times |E|$. Si la compatibilité est identique en toutes les paires de sites, indépendamment de leur localisation sur l'image, les matrices ψ_{st} seront toutes égales. La probabilité *a posteriori* s'écrit maintenant

$$P(\mathtt{X} = \mathtt{x} | \mathtt{Y} = \mathtt{y}) \propto \prod_{s \in S,\, t \in \Gamma_s} \psi_{st}(\mathtt{x}_s, \mathtt{x}_t) \prod_{s \in S} \phi(\mathtt{x}_s, \mathtt{y}_s).$$

Le choix des fonctions ψ_{st} permet d'imposer certaines propriétés, de régularité par exemple, au champ $\mathtt{X}$ et celui de ϕ caractérise l'attache aux données $\mathtt{y}$.

La méthode de Belief Propagation consiste à faire circuler des messages locaux entre sites voisins, permettant de propager des informations dans toute l'image. Un message est envoyé d'un site s à un site t voisin, concernant le descripteur $\mathtt{x}_t$ du site t. Ce message, noté m_{st}, est une distribution de probabilité sur les valeurs $\mathtt{x}_t$, il est donc représenté par un vecteur de longueur $|E|$. Au départ, les messages sont initialisés par des distributions uniformes sur E. Il existe deux versions de l'algorithme de Belief Propagation, correspondant à deux méthodes de mises à jour des messages : la version "somme-produit"

$$m_{st}(\mathtt{x}_t) = C \sum_{\mathtt{x}_s \in E} \left(\psi_{st}(\mathtt{x}_s, \mathtt{x}_t)\, \phi(\mathtt{x}_s, \mathtt{y}_s) \prod_{u \in \Gamma_s \setminus \{t\}} m_{us}(\mathtt{x}_s) \right)$$

et la version "max-produit"

$$m_{st}(\mathtt{x}_t) = C \max_{\mathtt{x}_s \in E} \left(\psi_{st}(\mathtt{x}_s, \mathtt{x}_t)\, \phi(\mathtt{x}_s, \mathtt{y}_s) \prod_{u \in \Gamma_s \setminus \{t\}} m_{us}(\mathtt{x}_s) \right)$$

où C est une constante de normalisation, telle que $\sum_{\mathtt{x}_t \in E} m_{st}(\mathtt{x}_t) = 1$. La figure (4.4) illustre les deux manières de calculer les messages.

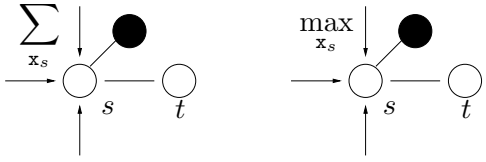

FIGURE 4.4: *Versions "somme-produit" et "max-produit" de mise à jour du message m_{st}.*

À partir des messages, on définit une croyance pour chaque site s, distribution de probabilité sur E. Elle est proportionnelle au produit de la fiabilité de l'observation avec tous les messages reçus au site s

$$b_s(\mathtt{x}_s) = C' \phi(\mathtt{x}_s, \mathtt{y}_s) \prod_{u \in \Gamma_s} m_{us}(\mathtt{x}_s)$$

où C' est une constante de normalisation, telle que $\sum_{\mathtt{x}_s \in E} b_s(\mathtt{x}_s) = 1$.

À chaque site, sont donc associés 4 messages destinés aux sites voisins et une croyance, tous sous forme de distributions de probabilité. L'algorithme dans sa version "max-produit", pour T itérations, est présenté sur la figure (4.5). À l'issue des T itérations, l'algorithme fournit donc une distribution de probabilité sur les valeurs de E en chaque site de l'image cherchée. On choisit alors pour un site s la valeur

$$\hat{\mathtt{x}}_s = \arg\max_{\mathtt{x}_s} \; b_s(\mathtt{x}_s).$$

1. Initialisation des messages

$$\forall s \in S,\ \forall t \in \Gamma_s,\ \forall \mathtt{x}_t \in E,\ \ m^0_{st}(\mathtt{x}_t) = \frac{1}{|E|}$$

2. Mise à jour des messages (T itérations)
   pour $i = 1 : T$

$$\forall s \in S,\ \forall t \in \Gamma_s,\ \forall \mathtt{x}_t \in E,$$
$$m^i_{st}(x_t) = C \max_{\mathtt{x}_s \in E} \left(\psi_{st}(\mathtt{x}_s, \mathtt{x}_t)\, \phi(\mathtt{x}_s, \mathtt{y}_s) \prod_{u \in \Gamma_s \setminus \{t\}} m^{i-1}_{us}(\mathtt{x}_s) \right)$$

3. Calcul des croyances

$$b_s(\mathtt{x}_s) = C' \phi(\mathtt{x}_s, \mathtt{y}_s) \prod_{u \in \Gamma_s} m^T_{us}(\mathtt{x}_s)$$

FIGURE 4.5: *Algorithme de Belief Propagation dans sa version "max-produit" pour T itérations.*

L'implémentation basique de l'algorithme, dans sa version "max-produit", a une complexité en $O(P|E|^2T)$ où P est le nombre de sites de l'image et T est le nombre de mises à jour des messages. Il est cependant possible de diminuer substanciellement le temps de calcul en combinant plusieurs techniques, notamment multi-échelle, non développées ici (elles sont décrites par Felzenszwalb et Huttenlocher dans [17]). Ceci est primordial lorsque l'on a un large ensemble de valeurs de descripteurs. Afin d'obtenir une convergence rapide, on peut aussi accélérer la mise à jour des messages. Dans [68], Tappen et Freeman proposent de propager les messages dans une direction et d'utiliser la mise à jour du site $s-1$ pour mettre à jour le site s. Ainsi, un site s peut transmettre une information à un site $s+k$ en une seule itération, au lieu de k pour la mise à jour synchrone.

4.1.4 Convergence de l'algorithme

L'algorithme de Belief Propagation donne de très bons résultats grâce à la puissance du passage des messages locaux, permettant de diffuser des informations d'un site à l'autre de l'image. Mais qu'en est-il de la convergence de l'algorithme ?

4.1.4.1 Version "somme-produit"

Pearl [55] puis Weiss [75] ont montré que pour un graphe simplement connexe de diamètre d, l'algorithme de Belief Propagation, dans sa version "somme-produit", converge en d itérations.

À la convergence, le vecteur de croyance b_s est égal à la distribution *a posteriori* marginale au site s, c'est-à-dire, en notant m^*_{st} les messages à la convergence,

$$\begin{aligned} b_s(\mathtt{x}_s) &= C\,\phi(\mathtt{x}_s,\mathtt{y}_s)\prod_{u\in\Gamma_s} m^*_{us}(\mathtt{x}_s) \\ &= \sum_{\mathtt{x}^s\in E^{|S|-1}} P(\mathtt{X}^s=\mathtt{x}^s,\mathtt{X}_s=\mathtt{x}_s|\mathtt{Y}=\mathtt{y}) \\ &= P(\mathtt{X}_s=\mathtt{x}_s|\mathtt{Y}=\mathtt{y}). \end{aligned}$$

On peut aussi calculer les distributions jointes *a posteriori* entre deux sites voisins

$$\begin{aligned} b_{st}(\mathtt{x}_s,\mathtt{x}_t) &= C\,\psi_{st}(\mathtt{x}_s,\mathtt{x}_t)\phi(\mathtt{x}_s,\mathtt{y}_s)\phi(\mathtt{x}_t,\mathtt{y}_t)\prod_{u\in\Gamma_s\setminus\{t\}} m^*_{us}(\mathtt{x}_s)\prod_{u\in\Gamma_t\setminus\{s\}} m^*_{ut}(\mathtt{x}_t) \\ &= P(\mathtt{X}_s=\mathtt{x}_s,\mathtt{X}_t=\mathtt{x}_t|\mathtt{Y}=\mathtt{y}). \end{aligned}$$

Pour les graphes quelconques, la convergence n'est pas assurée mais lorsqu'elle a lieu, les croyances b_s et b_{st} forment un point critique d'une approximation de l'énergie libre d'un système, appelée énergie libre de Bethe et définie par $E_{Bethe}\left(\{b_s,b_{st}\}\right) =$

$$\sum_{s,t\in S}\sum_{\mathtt{x}_s,\mathtt{x}_t\in E} b_{st}(\mathtt{x}_s,\mathtt{x}_t)\left(\ln b_{st}(\mathtt{x}_s,\mathtt{x}_t) - \ln\left(\psi_{st}(\mathtt{x}_s,\mathtt{x}_t)\,\phi(\mathtt{x}_s,\mathtt{y}_s)\,\phi(\mathtt{x}_t,\mathtt{y}_t)\right)\right)$$

$$-\sum_{s\in S}(q_s-1)\sum_{\mathtt{x}_s\in E} b_s(\mathtt{x}_s)\left(\ln b_s(\mathtt{x}_s) - \ln\phi(\mathtt{x}_s,\mathtt{y}_s)\right)$$

où q_s est le nombre de voisins du site s. Ce point est développé par Yedidia, Freeman et Weiss dans [78].

4.1.4.2 Version "max-produit"

Pour un graphe simplement connexe, la version "max-produit" de l'algorithme de Belief Propagation converge vers l'estimateur du maximum *a posteriori* (Pearl [55]). En effet, à la convergence,

$$b_s(\mathtt{x}_s) = C\,\max_{\mathtt{x}} P(\mathtt{X}=\mathtt{x}|\mathtt{X}_s=\mathtt{x}_s,\mathtt{Y}=\mathtt{y}).$$

La réalisation

$$\hat{\mathtt{x}} = \{\hat{\mathtt{x}}_s = \arg\max_{\mathtt{x}_s\in E} b_s(\mathtt{x}_s), s\in S\}$$

maximise la probabilité *a posteriori* $P(\mathtt{X}=\mathtt{x}|\mathtt{Y}=\mathtt{y})$. Remarquons qu'avec la version "somme-produit" de l'algorithme, on obtient les distributions marginales *a posteriori* en chaque site, mais la réalisation $\hat{x}$ qui en résulte ne maximise pas forcément la probabilité *a posteriori* sur le graphe.

Dans le cas d'un graphe quelconque, l'algorithme dans sa version "max-produit" ne converge pas toujours. La présence de boucles dans le graphe peut entraîner la circulation infinie de messages dans ces boucles, empêchant la convergence vers un équilibre stable. On observera plus loin des structures en damier, correspondant à l'oscillation d'un groupe de sites entre deux états. Cependant, lorsqu'il y a convergence, les résultats produits sont souvent assez bons, comme l'illustrent ceux obtenus par Levin, Zomet et Weiss pour la séparation de mouvements transparents [41] et les applications diverses (restauration, segmentation, super résolution...) présentées par Sharon dans [62]. Dans [76], Weiss et Freeman démontrent que si l'algorithme dans sa version "max-produit" converge dans un graphe avec boucles, la solution obtenue $\hat{\mathtt{x}}$ est le maximum de la probabilité *a posteriori* dans un voisinage particulier qu'ils appellent "Single Loops and Trees" (SLT). Le voisinage SLT d'une configuration $\mathtt{x}$ est défini comme l'ensemble des configurations ne différant de $\mathtt{x}$ que sur des combinaisons disjointes d'arbres et de boucles simples. Ainsi,

$$\forall \mathtt{x} \in SLT(\hat{\mathtt{x}}), \quad P(\mathtt{X} = \mathtt{x} | \mathtt{Y} = \mathtt{y}) < P(\mathtt{X} = \hat{\mathtt{x}} | \mathtt{Y} = \mathtt{y}).$$

4.1.5 Un exemple d'application de la Belief Propagation : la désoccultation

Pour illustrer les performances de la Belief Propagation, nous présentons quelques expériences de désoccultation d'images par application de la Belief Propagation. Ici, on restaure une image dont seul un faible pourcentage (10%) des niveaux de gris des pixels est connu. Pour cela, on utilise une image initiale quantifiée, ici en 32 niveaux de gris, dont on masque aléatoirement 90% des pixels. Les niveaux de gris de l'image quantifiée constituent l'ensemble E. La localisation des pixels manquants est connue et utilisée pour définir la fonction $\phi(\mathtt{x}_s, \mathtt{y}_s)$. Plus précisément, on possède des observations $\mathtt{y}_s$ sur certains sites s et on définit ϕ par

$$\forall x_s \in E, \;\; \phi(\mathtt{x}_s, \mathtt{y}_s) = \begin{cases} \mathbb{1}_{\mathtt{x}_s = \mathtt{y}_s} & \text{si } \mathtt{y}_s \text{ existe,} \\ \dfrac{1}{|E|} & \text{si } \mathtt{y}_s \text{ n'existe pas.} \end{cases}$$

La fonction de compatibilité ψ doit privilégier les niveaux de gris identiques ou proches pour des sites voisins. Elle est choisie commune à tous les sites de l'image et est donnée par

$$\psi(\mathtt{x}_s, \mathtt{x}_t) = (1 - e_p)\, e^{-\frac{|\mathtt{x}_s - \mathtt{x}_t|}{\sigma_p}} + e_p$$

où les paramètres e_p et σ_p permettent d'imposer une régularité plus ou moins forte. Cette fonction provient du modèle de variation totale de Osher, Rudin et Fatemi [54]. Ces derniers utilisent la fonction de potentiel $\rho(x) = |x|$ car elle préserve davantage de discontinuités que la fonction $\rho(x) = x^2$. Le choix de σ_p impose une régularité plus ou moins forte et le paramètre e_p, qui borne inférieurement les valeurs $\psi(\mathtt{x}_s, \mathtt{x}_t)$, autorise, s'il est important, davantage de discontinuités, correspondants aux contours. Les figures (4.6) et (4.7) présentent les résultats obtenus par application de la Belief Propagation dans sa version "max-produit" sur les images

La Cornouaille et Mandrille occultées à 90%, pour $e_p = 0$ ou 0.01 et $\sigma_p = 1$ ou 5. Bien que la transmission d'un message d'un bord de l'image à l'autre nécessite N itérations où N est la largeur de l'image, quelques itérations suffisent ici à atteindre un état stable. Ceci est dû à la dispersion uniforme des pixels dont le niveau de gris est connu dans l'image : ceux-ci conditionnent fortement les niveaux de gris des pixels voisins. Ainsi, très rapidement (une dizaine d'itérations), de larges plages de l'image sont remplies et leur niveau de gris n'évolue plus, comme illustré sur les deux figures (4.6) et (4.7). Le choix de e_p et σ_p est déterminant ; la valeur $e_p = 0.01$ permet d'obtenir des contours plus nets sur les images, c'est le cas de la coque et du mât du bateau sur la figure (4.6) et les résultats sont alors plus contrastés. Une valeur élevée de σ_p lisse l'image obtenue ; cet effet est visible sur la texture du pelage du mandrille sur la figure (4.7). Il n'y a pas de jeu de paramètres optimal pour toute désoccultation. Suivant la nature des images, on peut préférer une image bien segmentée, ou plus floue (par exemple pour images texturées). Ainsi, pour l'image La Cornouaille, présentant de grandes zones uniformes, les valeurs $e_p = 0.01$ et $\sigma_p = 5$ fournissent de bons résultats : à partir de l'image occultée à 90%, on identifie clairement le bateau, les mâts, les cordages et on distingue la bouée, l'homme à terre. Pour l'image Mandrille, plus texturée, les paramètres $e_p = 0$ et $\sigma_p = 1$ permettent de retrouver en partie les textures tout en reconnaissant les différentes parties de la tête, notamment les yeux et les narines alors que pour $\sigma_p = 5$, les zones texturées sont plus lissées.

Enfin, dans le cas d'occultations plus larges dans l'image, la Belief Propagation fournit aussi des résultats intéressants ; la version "somme-produit" est alors plus adaptée que la version "max-produit". En effet, la version "max-produit", en l'absence de données dans la zone occultée, a tendance à remplir la zone par un niveau de gris uniforme (ou des niveaux de gris très voisins), qui maximise la probabilité *a posteriori*. En revanche, la version "somme-produit", qui fait converger les distributions *a posteriori* marginales en chaque site, propage les données de sites en sites dans la zone occultée. Sur la figure (4.8), le détail de la bouche de Léna est occulté par une tâche de 60 pixels. Le résultat fourni par la Belief Propagation dans sa version "somme-produit" avec $e_p = 0.5$ et $\sigma_p = 1$ est tout-à-fait comparable à ceux obtenus sur la bouche de Léna par d'autres méthodes de désoccultation, comme l'interpolation jointe des niveaux de gris et des directions du gradient proposée par Ballester, Caselles et Verdera dans [2].

4.2 Application de la Belief Propagation à l'estimation de profondeurs

Le problème d'estimation des profondeurs d'une scène à partir de deux vues de cette scène, appelé aussi problème d'appariement stéréo, est un problème difficile à cause des occultations, du bruit sur les images, des discontinuités de profondeurs, des textures...

De nombreuses méthodes existent ; Scharstein et Szeliski en proposent une classification dans [60], basée sur les techniques mises en oeuvre lors des quatre principales étapes des algorithmes. La première est le calcul du coût d'appariement (réalisé en général par la différence absolue des

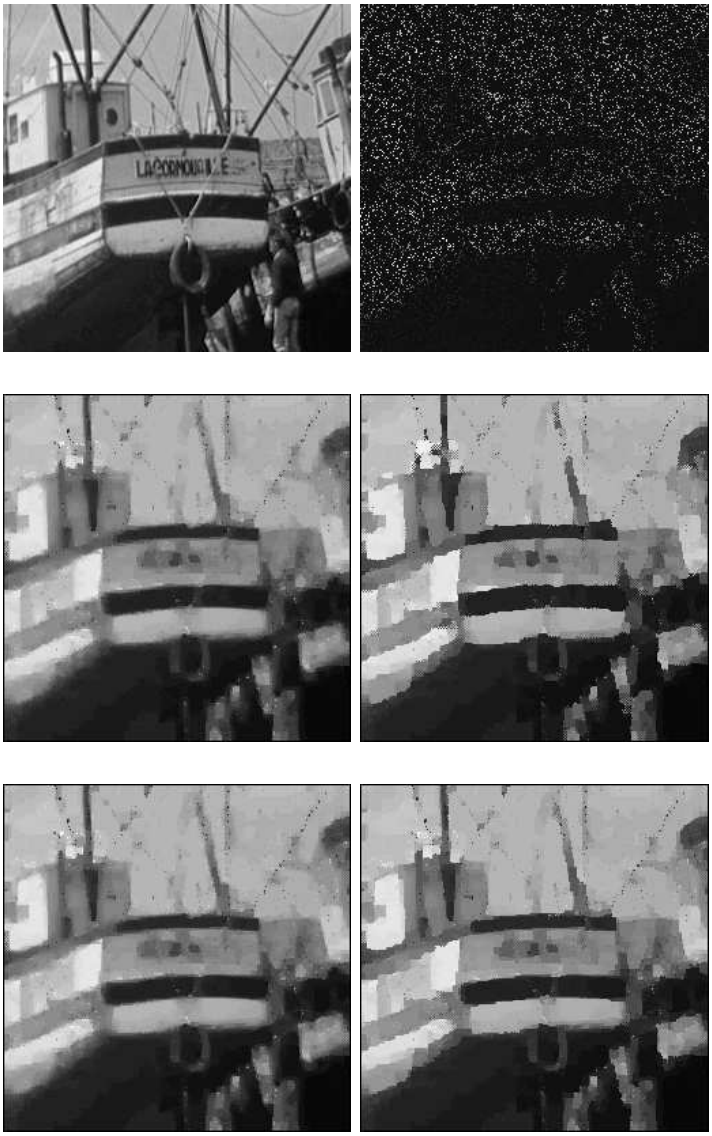

FIGURE 4.6: *Désoccultation par Belief Propagation de l'image La Cornouaille. En haut, l'image initiale et l'image masquée à* 90%*, puis les résultats de désoccultations obtenus après 20 itérations de l'algorithme dans sa version "max-produit", avec sur la colonne de gauche,* $e_p = 0$*, sur la colonne de droite,* $e_p = 0.01$ *et de haut en bas,* $\sigma_p = 1$ *et* 5*.*

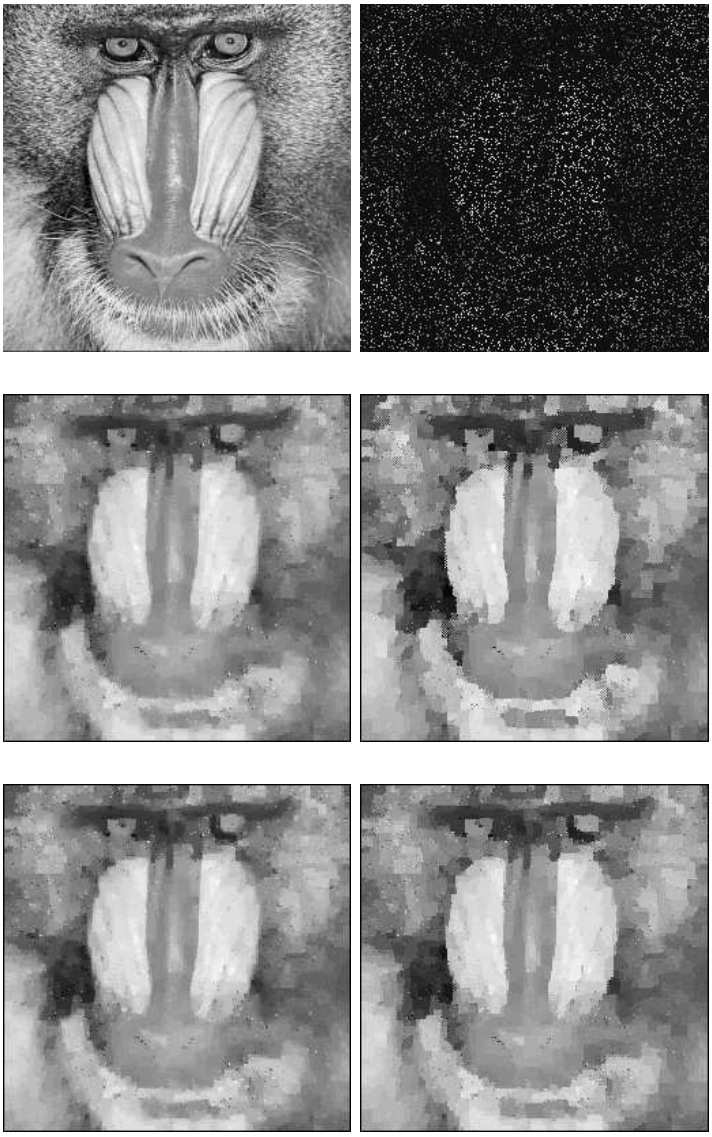

FIGURE 4.7: *Désoccultation par Belief Propagation de l'image Mandrille. En haut, l'image initiale et l'image masquée à* 90%*, puis les résultats de désoccultations obtenus après 20 itérations de l'algorithme dans sa version "max-produit", avec sur la colonne de gauche,* $e_p = 0$*, sur la colonne de droite,* $e_p = 0.01$ *et de haut en bas,* $\sigma_p = 1$ *et* 5*.*

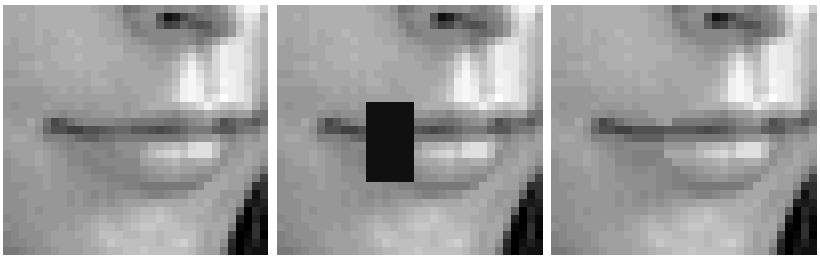

FIGURE 4.8: *Détail de la bouche de Léna. De gauche à droite, l'image originale quantifiée en* 32 *niveaux de gris, l'image occultée et le résultat donné par la Belief Propagation, dans sa version "somme-produit" avec* $e_p = 0.5$ *et* $\sigma_p = 1$.

intensités ou par la différence des intensités au carré), la deuxième le calcul du coût d'agrégation (somme des coûts sur une fenêtre par exemple), la troisième le calcul des profondeurs (en utilisant ou non une technique d'optimisation, comme la programmation dynamique, le recuit simulé, les Graph Cuts...) et la quatrième le raffinement post-traitement (on peut notamment utiliser un filtre médian pour supprimer les mauvais appariements de petite taille). Les méthodes mises en oeuvre à chaque étape sont comparées et les algorithmes testés sur une large base de données. Il ressort qu'un coût d'appariement robuste, c'est-à-dire tronqué, améliore les performances des algorithmes globaux. Il est cependant difficile de fixer le seuil au-dessus duquel on tronque les valeurs car la meilleure troncature varie avec chaque paire d'images. De façon générale, les meilleurs résultats sont obtenus lorsque l'on modélise l'image des profondeurs par un champ de Markov et que le problème d'inférence est résolu par des techniques de Graph Cuts ou de Belief Propagation. Les méthodes de diffusion et de coopération (inspirées par des stratégies de stéréovision humaine) fournissent également de bons résultats, sauf sur les bords, moins précis. Enfin, les méthodes locales sont les moins performantes.

Suite à cette évaluation, Tappen et Freeman ont comparé plus précisément dans [68] les méthodes de Graph Cuts et de Belief Propagation pour l'appariement stéréo. Les résultats obtenus sont très voisins. L'algorithme de Graph Cuts consiste à représenter le problème sous la forme d'un ensemble de noeuds et d'arêtes orientées pondérées et à réaliser une coupe du graphe de coût minimal, correspondant à la minimisation d'une énergie. Les Graph Cuts fournissent des solutions plus lisses, d'énergie plus faible que celles obtenues par la mise en oeuvre de la Belief Propagation mais ces solutions ne sont pas nécessairement plus proches des véritables profondeurs de la scène (car les solutions fournies par les Graph Cuts et la Belief Propagation ont toutes deux des énergies significativement plus faibles que celle du champ des véritables profondeurs). Enfin, le temps de calcul de la Belief Propagation dans sa version accélérée (utilisation immédiate de la mise à jour d'un message) et celui des Graph Cuts sont similaires.

Nous allons ici appliquer la méthode de Belief Propagation à l'estimation des profondeurs à partir de deux images consécutives dans une séquence et du mouvement de la caméra déterminé

par la méthode du chapitre 3. La différence avec les méthodes proposées dans [67, 60, 68] est l'application de la Belief Propagation à des images non rectifiées.

4.2.1 Présentation du problème

Dans le chapitre 3, nous avons estimé le mouvement de la caméra entre deux images consécutives dans une séquence en supposant que la profondeur Z de la scène filmée, la translation t de la caméra et la taille L des images vérifiaient, en unités de longueur focale

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} < 2\varepsilon,$$

avec $\varepsilon < 10^{-2}$, ce qui nous permettait de considérer la profondeur de la scène constante dans le repère de la caméra. La figure (4.9) présente deux images consécutives dans une séquence et le recalage de la deuxième sur la première grâce au mouvement de caméra estimé. On observe sur la superposition des images que certains objets de la scène sont légèrement décalés car leur profondeur diffère de la profondeur moyenne de la scène. On a observé des effets similaires sur les résultats de mosaïquage d'images éloignées dans la séquence, dans le chapitre 3. Nous allons utiliser ce décalage pour estimer un plan des profondeurs relatives de la scène dans le repère de la caméra avant son déplacement.

Revenons sur les formules exactes des correspondances des points entre deux images consécutives f et g, l'image g étant acquise après un mouvement de caméra (R, t) où R est une rotation de matrice

$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$$

et t est une translation de vecteur

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}.$$

Soient K et K' les domaines des plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ sur lesquels sont respectivement définies les images f et g. Soit (x, y) un point de K et (x', y') un point de K', deux points appariés, c'est-à-dire projections d'un même point de l'espace de profondeur $Z(x, y)$ dans le repère de la caméra avant le déplacement. Pour une longueur focale f_c, la formule suivante donne la relation entre (x, y), (x', y') et $Z(x, y)$

$$\begin{cases} x' = f_c \dfrac{a_1 x + a_2 y + f_c \, a_3 - f_c \, \langle \frac{t}{Z(x,y)}, R(i) \rangle}{c_1 x + c_2 y + f_c \, c_3 - f_c \, \langle \frac{t}{Z(x,y)}, R(k) \rangle} \\ \\ y' = f_c \dfrac{b_1 x + b_2 y + f_c \, b_3 - f_c \, \langle \frac{t}{Z(x,y)}, R(j) \rangle}{c_1 x + c_2 y + f_c \, c_3 - f_c \, \langle \frac{t}{Z(x,y)}, R(k) \rangle}. \end{cases}$$

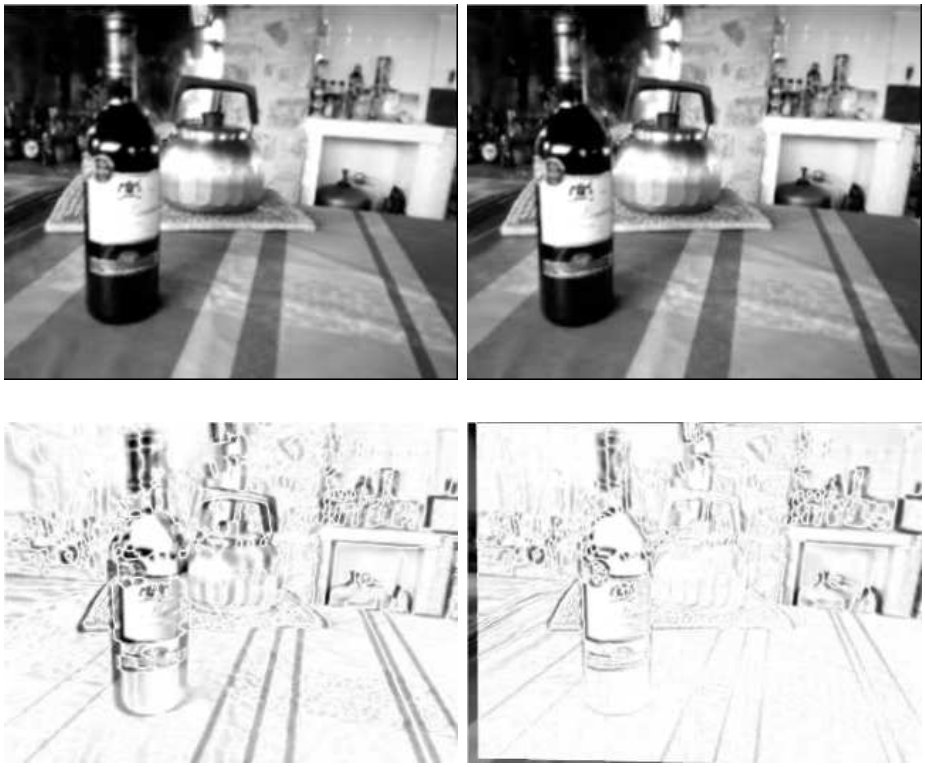

FIGURE 4.9: *En haut, deux images consécutives dans une séquence. En bas, à gauche la différence en valeur absolue entre les deux images et à droite la différence en valeur absolue entre la première image et la deuxième recalée sur la première avec le mouvement de caméra estimé par la méthode décrite dans le chapitre 3. Plus le niveau de gris est sombre et plus la différence est élevée. Le recalage est globalement bon sauf pour certaines profondeurs de la scène qui diffèrent trop de la profondeur moyenne : c'est le cas de la bouteille et de la zone du fond à droite.*

La méthode présentée dans le chapitre 3 estime une translation $\widetilde{t}$ égale à la translation divisée par une profondeur moyenne Z_0 de la scène. Nous allons maintenant, en utilisant le recalage des images, estimer en chaque point de K une distance à la profondeur Z_0, permettant d'obtenir une image des profondeurs relatives des objets. La formule précédente nous rappelle que dans les cas où il n'y a pas de translation, il est impossible de déduire la structure de la scène.

4.2.2 Disparités ou profondeurs ?

La plupart des méthodes qui estiment la structure d'une scène filmée à partir de deux images se placent dans le cas d'une caméra en translation horizontale : les plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ sont donc confondus. On dit alors que les images sont rectifiées, comme illustré sur la figure (4.10). Ces méthodes estiment non pas la profondeur de la scène mais la disparité d en chaque point de

K.

Définition 4.3 – *Soient (x, y) et (x', y') deux points appariés de deux images rectifiées. On appelle disparité au point (x, y) la différence*

$$d = y' - y.$$

Cette différence est liée à la profondeur $Z(x, y)$ du point projeté en (x, y) et (x', y') par

$$d = \frac{f_c}{Z(x, y)} d_{CC'}$$

où f_c est la longueur focale et $d_{CC'}$ la distance entre les centres optiques.

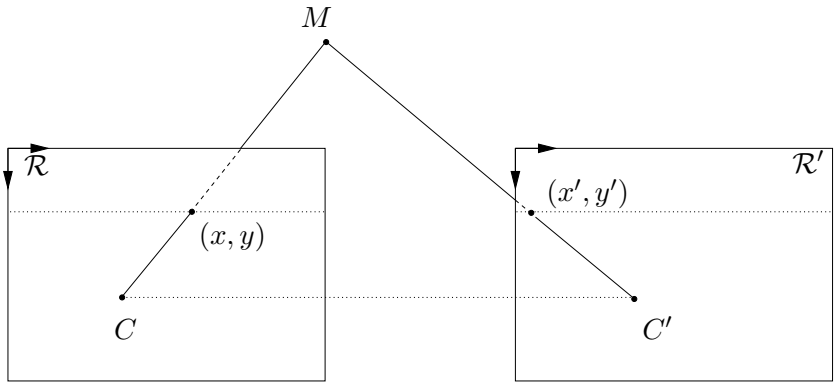

FIGURE 4.10: *Deux images rectifiées. Les points (x, y) et (x', y') des plans $\mathcal{R}$ et $\mathcal{R}'$ sont appariés ; comme la caméra est en translation horizontale, $x = x'$.*

Dans le cas d'un mouvement quelconque de caméra, on ne peut plus définir la disparité de la même façon.

Définition 4.4 – *Soit un mouvement de caméra différent d'une simple translation horizontale. Soient (x, y) et (x', y') deux points appariés dans les deux images obtenues avant et après le déplacement. Ces points sont deux projections d'un même point de l'espace 3D de profondeur $Z(x, y)$ dans le repère associé à la caméra avant son déplacement. On appelle disparité au point (x, y)*

$$d = \frac{1}{Z(x, y)}.$$

Dans la suite du document, nous ne donnerons qu'un exemple d'estimation de disparités dans le cas d'images rectifiées. Les autres résultats présenteront l'estimation de profondeurs.

4.2.3 Rectification

Comme mentionné précédemment, la plupart des méthodes estimant la structure d'une scène traitent seulement les cas de translation horizontale de la caméra [60]. Dans ce cas, des points appariés dans les deux images appartiennent à la même ligne horizontale. Ceci simplifie grandement le problème d'estimation de la structure et permet de développer des algorithmes très rapides, voire même temps réel [5, 22]. Si les mouvements de la caméra sont plus compliqués qu'une simple translation horizontale, ces méthodes appliquent un procédé aux images, appelé rectification, afin de se ramener au cas de la translation horizontale. Le procédé consiste à déterminer une transformation de chaque image telle que les lignes épipolaires conjuguées deviennent colinéaires et parallèles à l'axe horizontal des images [13, 18, 19].

En général, on projette les images sur un plan parallèle à la droite passant par les deux positions du centre optique (avant et après le mouvement de la caméra). Comme il existe une infinité de plans vérifiant cette condition, on peut choisir le plan minimisant la distorsion des images projetées ou plus simplement le plan parallèle à la droite d'intersection des plans rétiniens $\mathcal{R}$ et $\mathcal{R}'$ [13], comme illustré sur la figure (4.11).

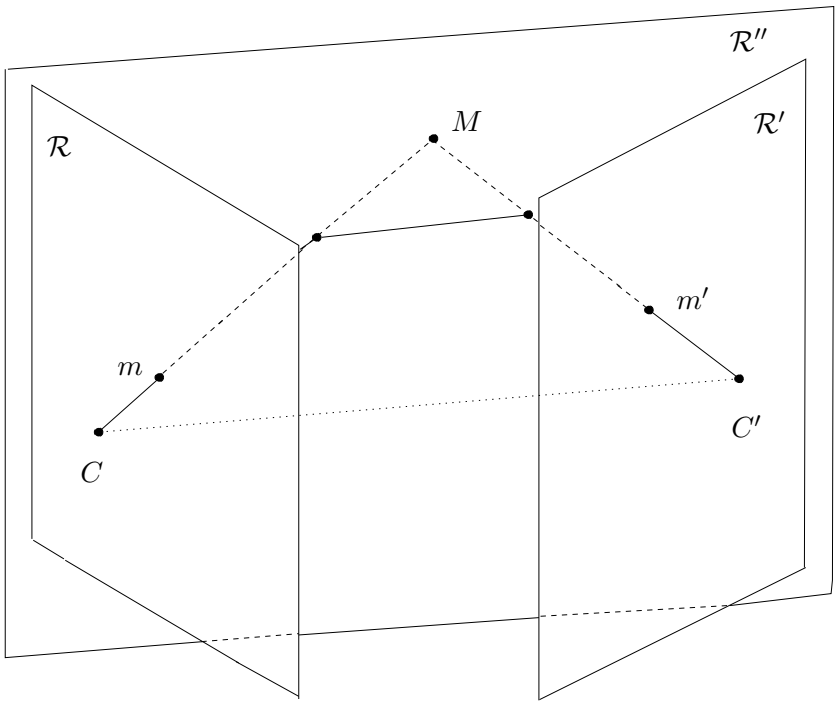

FIGURE 4.11: *Procédé de rectification. Les images des plans $\mathcal{R}$ et $\mathcal{R}'$ sont projetées sur le plan $\mathcal{R}''$ parallèle à la droite CC'. Ainsi, les lignes épipolaires deviennent parallèles à CC'.*

Cependant, bien que la rectification simplifie beaucoup le problème d'évaluation des disparités d'une scène, on peut s'interroger sur l'amélioration apportée par rapport à une méthode

utilisant directement les images non rectifiées. Schreer, Brandburg et Kauff présentent dans [61] une étude comparative entre l'estimation des disparités sur des vues rectifiées et non rectifiées. Dans les deux cas, la même approche hiérarchique de block-matching, décrite dans [52], est appliquée. La première comparaison de cette étude concerne la complexité. Lorsque les vues ne sont pas rectifiées, le calcul de la position des fenêtres candidates au block-matching nécessite l'approximation de la ligne épipolaire sur la grille discrète. Le coût de ce calcul est supérieur à deux fois celui de la rectification des images ; cependant, ce coût est marginal comparé à la complexité totale du procédé d'estimation des disparités : la différence de coût entre les deux méthodes n'est pas significative. La seconde comparaison s'attache aux résultats ; la méthode sur les images non rectifiées fournit de meilleurs résultats que ceux obtenus par la méthode sur les images rectifiées si la région d'intérêt couvre toute l'image, à cause de l'effet de distortion, induit par la rectification, plus important sur les bords de l'image. Pour la même raison, plus les directions des axes optiques diffèrent et plus il est préférable de travailler directement sur les images non rectifiées. Cependant, dans le cas d'objets bien segmentés au centre de l'image et d'axes optiques de directions voisines, les résultats obtenus par les deux méthodes sont très proches.

4.2.4 Utilisation de la Belief Propagation sur des images non rectifiées

4.2.4.1 Choix de la non rectification

Sun, Shum et Zheng proposent dans [67] l'application de l'algorithme de Belief Propagation au problème d'appariements stéréo pour des images rectifiées. Ici, nous choisissons de ne pas rectifier les images. La principale raison de ce choix est que nous ne disposons que d'une estimation du mouvement de la caméra et non d'une donnée exacte. De plus, les objets dans les images issues de séquences réelles que nous considérons occupent une très large partie des images (et souvent aussi les bords). Nous adaptons donc la méthode de Sun, Shum et Zheng au cas où le mouvement de la caméra est quelconque.

4.2.4.2 Description de la méthode

En utilisant le formalisme présenté dans les sections 4.1.2 et 4.1.3, on note $\mathtt{y} = \{f, g, \theta, \alpha, \beta, A, B, C\}$ les deux images consécutives et les paramètres du mouvement de la caméra estimé. On cherche à maximiser la probabilité *a posteriori* $P(\mathtt{X} = \mathtt{x}|\mathtt{Y} = \mathtt{y})$ où $\mathtt{x}$ est le champ de profondeurs de la scène du point de vue de l'image f. La matrice de compatibilité entre les sites a pour terme général

$$\psi(\mathtt{x}_s, \mathtt{x}_t) = (1 - e_p)\, e^{-\frac{|\mathtt{x}_s - \mathtt{x}_t|}{\sigma_p}} + e_p.$$

Cette fonction, utilisée aussi par Sun, Shum et Zheng dans [67], provient du modèle de variation totale de Osher, Rudin et Fatemi [54] comme nous l'avons déjà évoqué précédemment pour la

désoccultation. Le choix de σ_p impose une régularité plus ou moins forte et le paramètre e_p permet d'autoriser plus ou moins de discontinuités de profondeurs sur l'image.

Pour la fonction d'attache aux données ou vraisemblance locale en un site s, on choisit

$$\phi(\mathtt{x}_s, \mathtt{y}_s) = (1 - e_d)\, e^{-\dfrac{F(s, \mathtt{x}_s, \mathtt{y}_s)}{\sigma_d}} + e_d$$

avec

$$F(s, \mathtt{x}_s, \mathtt{y}_s) = |f(s) - g(s')|$$

où s' est le point de K' apparié au site s de K d'après le mouvement de la caméra estimé et la profondeur $\mathtt{x}_s$ donnée, c'est-à-dire, si $s = (x, y)$ et $s' = (x', y')$,

$$\begin{cases} x' = f_c \dfrac{a_1 x + a_2 y + f_c\, a_3 + f_c \frac{A}{\mathtt{x}_s}}{c_1 x + c_2 y + f_c\, c_3 + f_c \frac{C}{\mathtt{x}_s}} \\ \\ y' = f_c \dfrac{b_1 x + b_2 y + f_c\, b_3 + f_c \frac{B}{\mathtt{x}_s}}{c_1 x + c_2 y + f_c\, c_3 + f_c \frac{C}{\mathtt{x}_s}}. \end{cases} \tag{4.1}$$

La valeur $g(s')$ est calculée par interpolation bilinéaire, ce qui limite la sensibilité à l'échantillonnage. Les valeurs A, B, C sont les coordonnées de t/Z_0 dans la base $(R(i), R(j), R(k))$, Z_0 étant une profondeur moyenne de la scène ; en conséquence, les coordonnées $(A, B, C)/\mathtt{x}_s$ sont celles de $t/(Z_0\, \mathtt{x}_s)$ dans cette même base. Ainsi, la profondeur correspondant à $\mathtt{x}_s$ est égale à $Z_0\, \mathtt{x}_s$ dans le repère associé à la caméra avant son déplacement. Une profondeur retenue $\mathtt{x}_s$ inférieure à 1 signifie donc que l'objet représenté en ce site est placé en avant du plan de profondeur Z_0 et une profondeur $\mathtt{x}_s$ supérieure à 1 indique que l'objet se situe en arrière de ce plan.

4.2.4.3 Résultats

Dans le cas d'une translation horizontale de la caméra et donc d'images rectifiées, on retrouve les résultats donnés par Sun, Shum et Zheng dans [67], comme l'illustre la figure (4.12). Sur cette figure, on a estimé non pas les profondeurs mais les disparités, au sens de la définition 3, avec la version "max-produit" de la Belief Propagation ($e_p = 0.01$, $\sigma_p = 3.5$, $e_d = 0.05$, $\sigma_d = 20$).

On applique maintenant la méthode de Belief Propagation décrite ci-avant dans sa version "max-produit", à des images non rectifiées. L'ensemble des paramètres choisi est fixé : $e_p = 0.01$, $\sigma_p = 0.3$, $e_d = 0.05$ et $\sigma_d = 20$. Pour toutes les séquences utilisées, on suppose l'angle de vue égal à 120°.

Les figures (4.13), (4.14) et (4.15) présentent les résultats fournis par la méthode sur des images non rectifiées consécutives f et g et en utilisant une estimation du mouvement de caméra obtenue par la méthode du chapitre 3. Les profondeurs relatives appartiennent à l'ensemble de 16 éléments

$$E = \{0.45,\ 0.5,\ 0.55,\ 0.6,\ 0.65,\ 0.7,\ 0.75,\ 0.8,\ 0.9,\ 1,\ 1.1,\ 1.2,\ 1.4,\ 1.6,\ 1.8,\ 2\}\,.$$

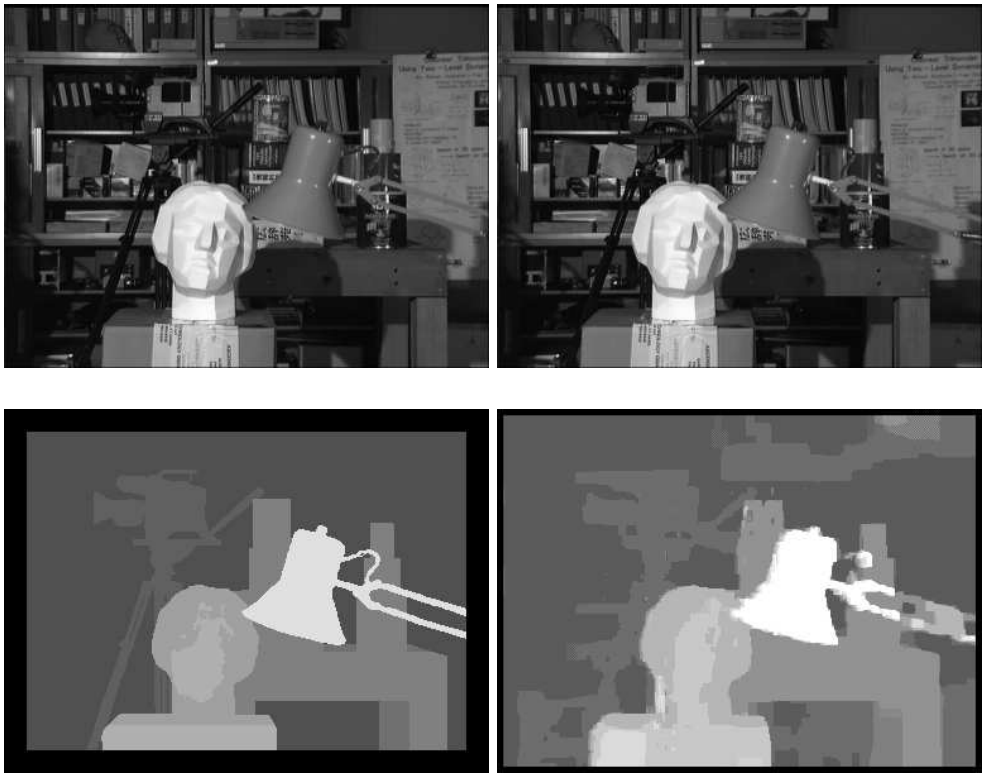

FIGURE 4.12: *En haut, deux images consécutives de la séquence Tsukuba. Au-dessous, à gauche le véritable champ des disparités et à droite l'estimation de ce champ obtenu par l'algorithme de Belief Propagation (100 itérations) connaissant le mouvement exact.*

Le choix de cet ensemble est déterminant ; on suppose que les objets se situent en avant ou en arrière d'une profondeur moyenne Z_0. Précisément, les valeurs $\mathtt{x}_s$ de E sont associées aux profondeurs $Z_0\,\mathtt{x}_s$, car la translation estimée est $\widetilde{t} = \frac{t}{Z_0}$. L'échantillonnage des profondeurs n'est pas régulier car les effets des profondeurs sur les translations ne sont pas linéaires : une profondeur égale à $1-\epsilon$ $(0 < \epsilon < 1)$ entraîne une variation dans la translation sur l'image d'amplitude supérieure à celle générée par une profondeur égale à $1+\epsilon$, soit

$$\frac{\widetilde{t}}{1-\epsilon} - \widetilde{t} > \widetilde{t} - \frac{\widetilde{t}}{1+\epsilon}.$$

En utilisant la formule (4.1), on peut calculer d'une part, l'image g recalée sur f en ne tenant compte que du mouvement de caméra estimé (c'est-à-dire avec $\mathtt{x}_s = 1$ pour tous les sites) et d'autre part, l'image g recalée sur f avec le mouvement de caméra et les profondeurs de la scène estimées sur l'image f. Les normes L^1 moyennes des différences entre f et les images g recalées sont données dans le tableau (4.1). Comme attendu, la norme des différences diminue significativement lorsque les profondeurs sont utilisées pour le recalage.

	Différence moyenne en norme L^1 entre f et g recalée avec	
	le mouvement de la caméra	le mouvement de la caméra + les profondeurs
Figure (4.13)	20.60	9.92
Figure (4.14)	7.99	1.86
Figure (4.15)	15.38	7.07

TABLEAU 4.1: *Norme L^1 (moyennée sur le nombre de pixels) des images de différence entre l'image f et les images g recalées pour les figures (4.13), (4.14) et (4.15).*

Sur la figure (4.13), la scène est constituée d'un arbre au premier plan, d'un ensemble de maisons et du ciel au dernier plan, d'un terrain en pente entre les deux. L'application de la Belief Propagation, utilisant le mouvement estimé entre les deux images consécutives, permet de bien détecter l'arbre au premier plan, à peu près correctement le terrain en pente et le ciel. Les profondeurs des maisons sont estimées égales à celles du ciel (ceci dépend de l'ensemble des profondeurs choisies). Le plan est simplifié mais clair.

Sur la figure (4.14), les objets sont plus rapprochés ; l'ensemble E utilisé est alors moins adapté au couple d'images (mais on choisit de n'utiliser aucune connaissance *a priori* sur la structure de la scène). Trois plans sont seulement détectés : le premier plan avec le mannequin, le second plan composé des différents objets et le fond. Les contours des objets du second plan sont peu précis, notamment car leur ombre leur est associée et accolée lors de l'estimation de profondeurs (sur la carafe et le pot de fleurs par exemple).

FIGURE 4.13: *En haut, deux images consécutives extraites de la séquence Flower Garden. Puis, l'estimation des profondeurs par la Belief Propagation (*100 *itérations) en utilisant le mouvement de la caméra entre les deux images estimé par la méthode du chapitre 3.*

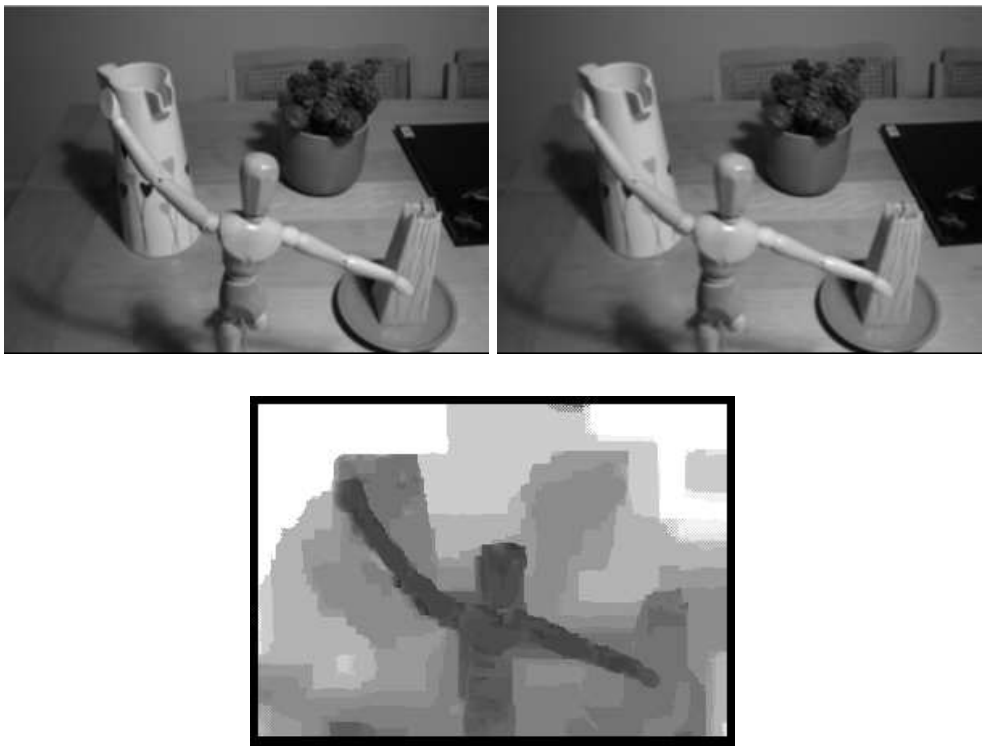

FIGURE 4.14: *En haut, deux images consécutives extraites d'une séquence. Puis, l'estimation des profondeurs par la Belief Propagation (*100 *itérations) en utilisant le mouvement de la caméra estimé dans le chapitre 3.*

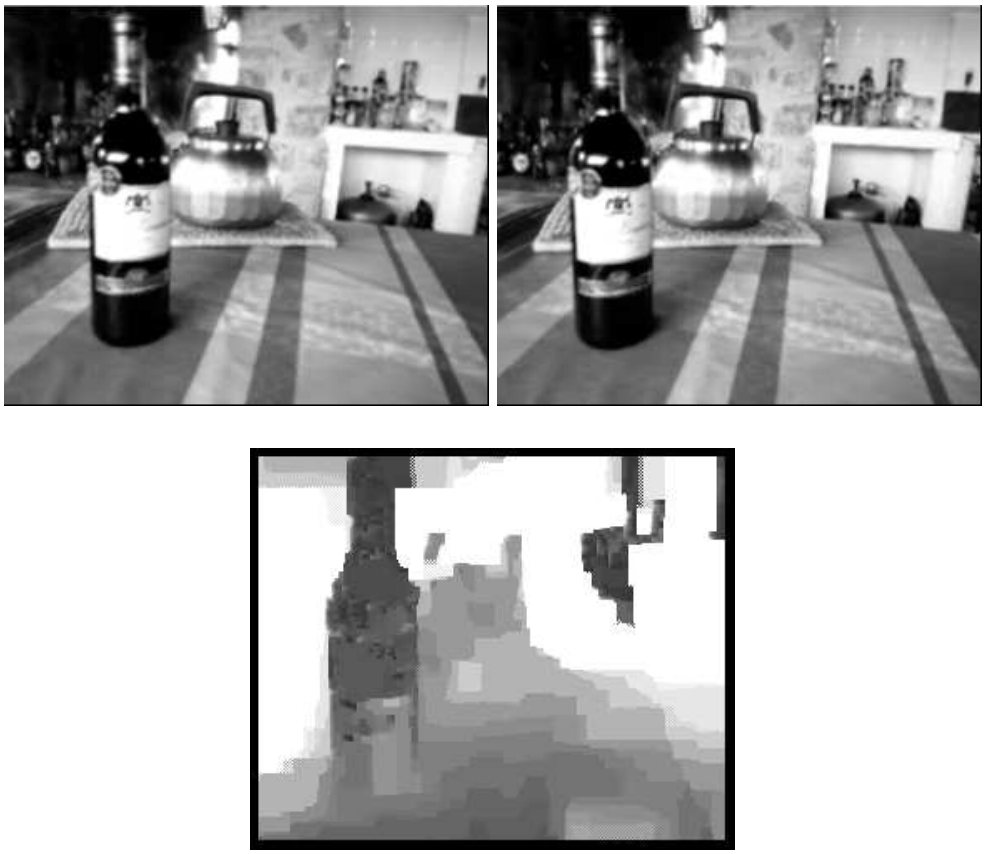

FIGURE 4.15: *En haut, deux images consécutives extraites d'une séquence. Puis, l'estimation des profondeurs par la Belief Propagation (*100 *itérations) en utilisant le mouvement de la caméra estimé dans le chapitre 3.*

Les images utilisées pour l'estimation des profondeurs sur la figure (4.15) sont plus complexes car la luminosité changeante modifie les ombres et les effets de réverbération entre les images consécutives ; ces effets sont notamment visibles sur la bouteille et la bouilloire. Le résultat est assez grossier ; on identifie la bouteille au premier plan et le corps de la bouilloire (mais pas son anse) au second plan. Un dégradé de profondeurs est observé sur une partie de la nappe. Cependant, le rebord de la table est très imprécis, et une zone de l'image en haut à droite est détectée au premier plan alors qu'elle appartient au fond.

En conclusion, les résultats d'estimation de profondeurs obtenus ne sont pas toujours satisfaisants. Pour des scènes simples, on obtient des plans simplifiés mais clairs de la structure mais pour des scènes plus compliquées, comme sur la figure (4.15), les résultats sont plus grossiers. Ceci est principalement dû à l'utilisation d'un mouvement de caméra non pas exact mais estimé ; en effet, l'estimation des profondeurs dépend étroitement de la précision de la décomposition du mouvement en une rotation et une translation. Pour améliorer l'estimation du mouvement et par conséquent celle des profondeurs, on propose d'itérer le procédé.

4.3 Estimation itérative des profondeurs et du mouvement de caméra

4.3.1 Description

On propose un algorithme itératif visant à améliorer à la fois l'estimation du mouvement de la caméra et celle des profondeurs. L'idée est d'utiliser les estimations des profondeurs Z pour estimer des mouvements 2D sur des zones de l'image de profondeurs voisines et en déduire une nouvelle estimation du mouvement de la caméra. À partir du nouveau mouvement, on estime les profondeurs, etc.

Plus formellement, on réalise une partition de l'ensemble E des profondeurs en H intervalles $I_1, \ldots, I_H$ de profondeurs moyennes $z_1, \ldots, z_H$. À partir de l'estimation du mouvement de la caméra par la méthode décrite dans le chapitre 3, on estime les profondeurs par Belief Propagation. Les intervalles $I_1, \ldots, I_h$ définissent alors une partition de l'image des profondeurs obtenue, soit du domaine K. Pour h appartenant à $\{1, \ldots, H\}$, on estime un mouvement $2D$ entre f et g en ne considérant que les points de K de profondeur estimée appartenant à I_h. Ainsi, on obtient H mouvements 2D, associés chacun à un ensemble de profondeurs. Ces H mouvements $2D$ conduisent à une estimation du mouvement de la caméra, par les calculs présentés ci-après. Rappelons d'abord que pour tout couple de points appariés (x, y) et (x', y') de K et K', projections d'un point de profondeur $Z(x, y)$ dans le repère de la caméra avant le déplacement, le flot optique est calculé par le logiciel Motion2D suivant le modèle

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix}$$

et la correspondance avec les six paramètres du mouvement $(\theta, \alpha, \beta, A, B, C)$ est la suivante

$$\begin{cases} c_1 = f_c \left(\dfrac{A}{Z(x,y)} - \alpha \sin\theta \right) & c_2 = f_c \left(\dfrac{B}{Z(x,y)} + \alpha \cos\theta \right) \\ a_1 = -\dfrac{C}{Z(x,y)} & a_2 = \beta \\ q_1 = -\dfrac{\alpha}{f_c} \sin\theta & q_2 = \dfrac{\alpha}{f_c} \cos\theta. \end{cases}$$

À partir de cette correspondance et des mouvements 2D estimés sur les H supports de profondeurs,

$$\left\{ \left(c_1^h, c_2^h, a_1^h, a_2^h, q_1^h, q_2^h \right),\ 1 \leq h \leq H \right\},$$

on calcule les six paramètres du mouvement de caméra correspondant. Pour cela, on pondère chacune des estimations par p_h, fraction de la surface de K de profondeur appartenant à I_h. Ainsi,

$$\theta = \begin{cases} -\arctan\left(\sum_{h=1}^H p_h\ q_1^h / \sum_{h=1}^H p_h\ q_2^h \right) & \text{si } \sum_{h=1}^H p_h\ q_2^h > 0 \\ -\arctan\left(\sum_{h=1}^H p_h\ q_1^h / \sum_{h=1}^H p_h\ q_2^h \right) + \pi & \text{si } \sum_{h=1}^H p_h\ q_2^h < 0 \\ \pi/2 & \text{si } \sum_{h=1}^H p_h\ q_2^h = 0 \text{ et } \sum_{h=1}^H p_h\ q_1^h > 0 \\ -\pi/2 & \text{si } \sum_{h=1}^H p_h\ q_2^h = 0 \text{ et } \sum_{h=1}^H p_h\ q_1^h \leq 0, \end{cases}$$

$$\alpha = f_c \sqrt{\left(\sum_{h=1}^H p_h\ q_1^h \right)^2 + \left(\sum_{h=1}^H p_h\ q_2^h \right)^2} \qquad \text{et}$$

$$\beta = \sum_{h=1}^H p_h\ a_2^h. \tag{4.2}$$

Le calcul de A, B et C tient compte des profondeurs moyennes des intervalles I_h,

$$A = \sum_{h=1}^{H} \left(\frac{c_1^h}{f_c} + \alpha \sin\theta \right) p_h \; z_h$$

$$B = \sum_{h=1}^{H} \left(\frac{c_2^h}{f_c} - \alpha \cos\theta \right) p_h \; z_h \tag{4.3}$$

$$C = - \sum_{h=1}^{H} p_h \; a_1^h \; z_h.$$

Les étapes de l'algorithme sont présentées sur la figure (4.16). La fonction de profondeur Z_i estimée à l'itération i est définie sur K. On note D_i les six paramètres du mouvement estimé à l'itération i, $D_i = (\theta_i, \alpha_i, \beta_i, A_i, B_i, C_i)$, et ψ_{Z_i,D_i} la fonction associant à un point (x, y) de K, le point (x', y') apparié de K', à partir du mouvement D_i et des profondeurs Z_i estimés, c'est-à-dire $(x', y') = \psi_{Z_i,D_i}(x, y)$ avec

$$\begin{cases} x' = f_c \dfrac{a_1^i x + a_2^i y + f_c \; a_3^i + f_c \frac{A_i}{Z_i(x,y)}}{c_1^i x + c_2^i y + f_c \; c_3^i + f_c \frac{C_i}{Z_i(x,y)}} \\ \\ y' = f_c \dfrac{b_1^i x + b_2^i y + f_c \; b_3^i + f_c \frac{B_i}{Z_i(x,y)}}{c_1^i x + c_2^i y + f_c \; c_3^i + f_c \frac{C_i}{Z_i(x,y)}}, \end{cases}$$

où $\begin{pmatrix} a_1^i & a_2^i & a_3^i \\ b_1^i & b_2^i & b_3^i \\ c_1^i & c_2^i & c_3^i \end{pmatrix}$ est la matrice de rotation associée aux angles $(\theta_i, \alpha_i, \beta_i)$. La partition de E et le nombre d'itérations N sont préalablement fixés. Le couple mouvement/profondeurs retenu est celui minimisant la différence en norme L_1 entre l'image f et l'image g recalée sur f avec les profondeurs et le mouvement de la caméra estimés lors des N itérations.

4.3.2 Résultats et discussion

On présente les résultats de l'algorithme sur les figures (4.17) et (4.18). L'ensemble des profondeurs E utilisé pour l'estimation et les paramètres de l'algorithme de Belief Propagation sont inchangés. On choisit la partition de $E = I_1 \cup I_2 \cup I_3$ avec

$$\begin{cases} I_1 = \{0.45,\ 0.5, \ldots, 0.75\} & \text{de moyenne } z_1 = 0.6 \\ I_2 = \{0.8, 0.9, \ldots, 1.2\} & \text{de moyenne } z_2 = 1 \\ I_3 = \{1.4, 1.6, \ldots, 2\} & \text{de moyenne } z_3 = 1.7. \end{cases}$$

Le mouvement sera donc indépendamment estimé sur chacun des trois plans de profondeurs. On utilise seulement trois plans pour limiter les erreurs d'estimation. En effet, l'estimation initiale

1. Initialisation
   - estimation du mouvement D_1 entre f et g
   - estimation de Z_1 à partir de f, g et D_1 par Belief Propagation
   - $\widehat{Z} = Z_1$, $\widehat{D} = D_1$
2. Pour $i = 2 \dots N$
   - pour $h = 1 \dots H$
     - estimation du mouvement 2D $(c_1^h, c_2^h, a_1^h, a_2^h, q_1^h, q_2^h)$ sur $K \cap \{Z_i \in I_h\}$
     - calcul du poids $p_h = \dfrac{\#\{Z_i \in I_h\}}{\#K}$
   - calcul de $D_i = (\theta_i, \alpha_i, \beta_i, A_i, B_i, C_i)$ à partir des H mouvements $2D$ et des H poids selon les formules (4.2) et (4.3)
   - estimation de Z_i à partir de f, g et D_i par Belief Propagation
   - Si $\|g \circ \psi_{Z_i, D_i} - f\|_1 < \|g \circ \psi_{\widehat{Z}, \widehat{D}} - f\|_1$ alors

$$\widehat{Z} = Z_i, \qquad \widehat{D} = D_i$$

FIGURE 4.16: *Algorithme itératif (N itérations) d'estimation du mouvement de la caméra et des profondeurs de la scène, une partition $I_1 \cup \dots \cup I_H$ de l'ensemble des profondeurs E ayant été préalablement définie. Les intervalles $I_1, \dots, I_H$ ont pour profondeurs moyennes $z_1, \dots, z_H$.*

des profondeurs peut être assez fortement erronée ; dans ce cas, plus on considérera de plans de profondeurs et plus l'erreur d'estimation sur le nouveau mouvement de caméra sera importante.

La figure (4.17) rappelle le résultat précédent et montre le résultat obtenu pour l'estimation des profondeurs après 15 itérations de l'algorithme présenté sur la figure (4.16). Le résultat obtenu en itérant successivement les estimations de mouvement et de profondeurs est nettement meilleur que celui obtenu seulement avec la première estimation du mouvement : on identifie clairement la bouteille, la bouilloire (avec son anse cette fois) et le dégradé de profondeurs de la table. Seule la partie à gauche de la bouteille est mal localisée au fond de la scène. Malgré cela, le bord de la table est correctement détecté.

Sur la figure (4.18), le résultat initial obtenu à partir du mouvement estimé entre les deux images permet d'identifier le parasol (et quelques éléments du réverbère) au premier plan, les façades d'immeubles à droite et à gauche au second plan et le fond entre les deux bâtiments. Cependant, sur le bâtiment de gauche, certaines fenêtres sont faussement détectées en avant de la scène et sur le bâtiment de droite, de profondeur uniforme (car l'axe optique lui est orthogonal), on observe plusieurs niveaux de profondeurs. Après 15 itérations de l'algorithme, le résultat

FIGURE 4.17: *En haut, les deux images consécutives à partir desquelles on estime les profondeurs. Au milieu, l'estimation des profondeurs par la Belief Propagation (100 itérations) en utilisant seulement le mouvement de la caméra estimé. En bas, le résultat obtenu pour 15 itérations de l'algorithme présenté sur la figure (4.16) (avec chaque fois 100 itérations de la Belief Propagation). Les normes L^1 moyennes des différences entre l'image f et l'image g recalée sur f avec le mouvement et les profondeurs estimées sont respectivement égales à* 7.07 *pour l'image des profondeurs du milieu et* 5.03 *pour l'image du bas.*

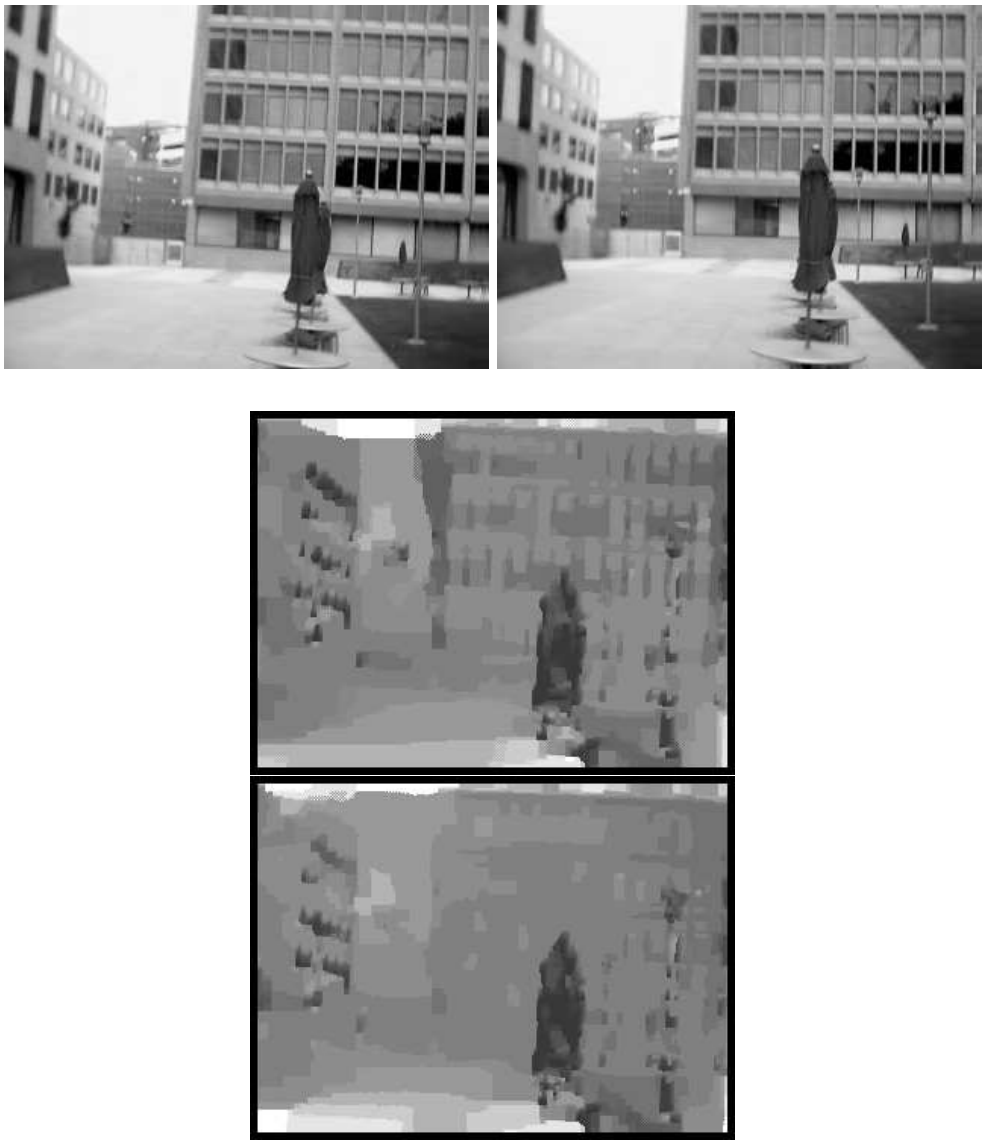

FIGURE 4.18: *En haut, les deux images consécutives à partir desquelles on estime les profondeurs. Au milieu, l'estimation des profondeurs par la Belief Propagation (100 itérations) en utilisant seulement le mouvement de la caméra estimé. En bas, le résultat obtenu pour 15 itérations de l'algorithme présenté sur la figure (4.16) (avec chaque fois 100 itérations de la Belief Propagation). Les normes L^1 moyennes des différences entre l'image f et l'image g recalée sur f avec le mouvement et les profondeurs estimées sont respectivement égales à* 5.07 *pour l'image des profondeurs du milieu et* 4.43 *pour l'image du bas.*

est très proche de la carte des profondeurs initiale mais les deux effets sur les profondeurs des bâtiments mentionnés ci-avant sont atténués.

L'algorithme proposé est purement heuristique ; la qualité des cartes de profondeurs estimées est calculée par la différence en norme L^1 entre l'image f et l'image g recalée sur f avec le mouvement et les profondeurs estimées. Pour tous les résultats présentés, nous avons utilisé le même ensemble de profondeurs E ; on ne suppose alors aucune connaissance *a priori* sur la structure de la scène. Malgré cette restriction, nous obtenons des résultats qui, sans être parfaits, donnent une bonne information de la structure de la scène, aussi bien pour une scène d'intérieur (figure (4.17)) avec des objets assez proches les uns des autres, que pour une scène d'extérieur où les composantes sont beaucoup plus éloignées (figure (4.18)). Ces résultats améliorent (plus ou moins nettement) la qualité des images de profondeurs obtenues en utilisant seulement la première estimation du mouvement de caméra. Notons qu'il est nécessaire que cette première estimation du mouvement ne soit pas trop erronée car en dépend l'estimation première des profondeurs, sur laquelle sont basées les estimations des mouvements suivants, etc.

De plus, l'estimation du mouvement sur différents plans de profondeurs permet d'étendre les limites du cadre défini dans le chapitre 3 pour l'estimation du mouvement de la caméra. Il est maintenant possible d'estimer correctement le mouvement pour des scènes présentant des variations de profondeurs plus importantes que celles autorisées dans le chapitre 3 c'est-à-dire telles que

$$\left(\frac{1}{Z_{inf}} - \frac{1}{Z_{sup}}\right) \|t\| \, (L+1) \, G_{max} < 2\varepsilon$$

pour $\varepsilon = 10^{-2}$. Il suffit que chaque plan de profondeur défini par la partition (et non toute la scène) vérifie cette dernière condition pour que chaque estimation soit correcte, sous réserve que la première estimation du mouvement ne soit pas trop faussée par de grandes variations de profondeurs. En revanche, l'algorithme n'est pas adapté à l'estimation de translations plus importantes que celles observées entre deux images consécutives dans une séquence car nous n'avons pas intégré dans la Belief Propagation de traitement particulier pour les occultations apparaissant dans le cas de translations conséquentes ; les profondeurs sont alors mal estimées car chaque point de l'image f est associé à un point de l'image g.

4.4 Conclusion

Dans ce chapitre, nous avons utilisé le mouvement de la caméra estimé entre deux images consécutives d'une séquence pour déterminer un plan de profondeurs de la scène 3D filmée. Nous avons appliqué pour cela un algorithme probabiliste convergeant vers la configuration des profondeurs maximisant la probabilité *a posteriori* connaissant le mouvement estimé et les deux images. L'ensemble des profondeurs testées et les paramètres de l'algorithme étant fixés, on obtient des résultats plus ou moins satisfaisants, suivant la structure de la scène et l'erreur initiale sur l'estimation du mouvement.

Pour améliorer ces résultats, on itère le procédé en estimant le mouvement 2D non plus directement entre les deux images mais sur des zones de profondeurs estimées voisines. En tenant compte des profondeurs moyennes de ces zones, on calcule, à partir des nouvelles estimations de mouvements 2D, un nouveau mouvement de caméra dans l'espace, utilisé par la Belief Propagation, etc. Les résultats sur les profondeurs sont visiblement meilleurs. La méthode peut aussi permettre d'estimer plus finement le mouvement de la caméra, notamment lorsque les variations des inverses des profondeurs sont importantes ; le cadre d'application de la méthode du chapitre 3 peut être élargi.

Chapitre 5

Sur l'injectivité du flot optique

Dans ce chapitre, nous étudions en détail et d'un point de vue théorique l'injectivité de l'application qui associe un flot optique au film d'une scène statique, c'est-à-dire à un mouvement de caméra et à la surface filmée. Nous prouvons que cette application est injective si le flot est observé, dans le cas d'une projection sténopé, sur le plan $\{Z = 1\}$ tout entier. Nous avons appris postérieurement que ce résultat avait déjà été obtenu par Brodsky, Fermüller et Aloimonos dans [7], par une démonstration différente. De plus, à partir de deux mouvements de caméra, nous décrivons le domaine d'observation du plan où les flots générés sont susceptibles d'être identiques et donnons les équations des surfaces filmées qui, associées à ces deux mouvements de caméra conduisent au même flot optique. Nous retrouvons alors les résultats de Horn [31] et Maybank [47] sur la nature de ces surfaces.

5.1 Présentation du problème

Le problème de l'estimation du mouvement de la caméra et de la structure de la scène à partir du flot optique, quand la caméra évolue dans un environnement statique, a été et est toujours très étudié. Cependant, un aspect théorique important du problème est habituellement peu considéré ; la plupart des auteurs présentent des méthodes en supposant qu'à un flot optique donné correspond exactement un mouvement de caméra et une surface filmée. Ce n'est pas toujours le cas. Par exemple, un flot optique généré par une caméra filmant une surface plane est toujours ambigu ; il existe une autre surface plane et un autre mouvement de l'observateur produisant le même flot, comme montré dans [43, 66, 46]. Dans le cas général, Horn [31] et Maybank [47] ont montré qu'un flot optique ne peut être ambigu que si les surfaces filmées appartiennent à une classe de surfaces particulières : les hyperboloïdes à une nappe, vues par un point de leur surface.

Dans ce chapitre, nous étudierons les flots optiques ambigus, du point de vue du domaine où le flot est observé. Pour cela, nous nous placerons dans le cadre théorique suivant : nous supposerons que le mouvement de la caméra est continu dans le temps, ce qui implique que

nous travaillerons avec des algèbres de Lie plutôt que des groupes et nous considérerons non plus la projection sténopé mais la projection stéréographique qui, en toute généralité, simplifie nos démonstrations. Nous commencerons donc par quelques rappels sur les mouvements infinitésimaux et nous introduirons des notations, la projection stéréographique et les formules liant flot optique, mouvement de caméra et profondeurs de la scène. Nous démontrerons au passage quelques résultats théoriques de projections correspondant à des morphismes d'algèbres de Lie. Puis, nous étudierons l'injectivité de la fonction qui associe un flot optique à un mouvement de caméra et à un plan des profondeurs de la scène filmée. On montre que cette application est injective si le domaine d'observation du flot est le disque unité du plan $\{Z = 0\}$ (car on considère la projection sur la sphère unité suivie de la projection stéréographique), ce qui revient à démontrer l'injectivité sur le plan $\{Z = 1\}$ tout entier pour la projection sténopé. Brodsky, Fermüller et Aloimonos ont obtenu ce résultat dans [7] en analysant le flot optique directement sur la sphère unité. Enfin, à partir de deux mouvements infinitésimaux de caméra, on décrira le domaine d'observation sur lequel les flots optiques générés peuvent être égaux, et les surfaces filmées qui, associées aux deux mouvements donnés, conduiront à un même flot optique.

5.2 Mouvement de caméra, profondeurs et flot optique

5.2.1 Mouvement de caméra et champ de vecteurs dans $\mathbb{R}^3$

On considère une caméra en mouvement dans un environnement statique. Ceci revient aussi à considérer l'environnement en mouvement dans le repère de la caméra. Pour décrire cette situation, rappelons que l'on note $SE(3)$ le groupe des déplacements rigides de $\mathbb{R}^3$, comme présenté dans le chapitre 1. Ce groupe est l'ensemble des matrices

$$\left\{ \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \text{ où } R \in SO(3),\ t \in \mathbb{R}^3 \right\}.$$

muni de la multiplication matricielle. Il agit sur l'hyperplan $\{T = 1\}$ de $\mathbb{R}^4$. On notera $M = (X, Y, Z)$ les coordonnées euclidiennes d'un point de $\mathbb{R}^3$ et $\mathbf{M} = (X, Y, Z, 1)$ ses coordonnées dans l'hyperplan.

L'algèbre de Lie du groupe $SE(3)$ est notée $\mathfrak{se}(3)$; c'est l'ensemble des matrices

$$\left\{ \begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3) \in \mathbb{R}^6 \right\}.$$

muni de l'addition matricielle et du crochet de Lie des matrices. Dans la suite, on écrira plus brièvement les éléments de $\mathfrak{se}(3)$

$$g = (\omega, v) = \begin{pmatrix} [\omega]_\times & v \\ 0 & 0 \end{pmatrix}.$$

Proposition 5.1 – *Soit $g = (\omega, v)$ appartenant à $\mathfrak{se}(3)$. Pour tout réel t, on a*

$$\exp(tg) = \begin{cases} \begin{pmatrix} e^{t[\omega]_\times} & \frac{1}{\|\omega\|^2}\left((I_3 - e^{t[\omega]_\times})\,[\omega]_\times + t\,\omega\omega^T\right) v \\ 0 & 1 \end{pmatrix} & si\ \omega \neq 0 \\ \begin{pmatrix} I_3 & t\,v \\ 0 & 1 \end{pmatrix} & sinon. \end{cases}$$

Les éléments de $\mathfrak{se}(3)$ produisent des champs de vecteurs dans $\mathbb{R}^3$. En effet, considérons g appartenant à $\mathfrak{se}(3)$ et un point M de $\mathbb{R}^3$. Alors $\exp(tg)$ est un déplacement de $SE(3)$,

$$\mathbf{M}(t) = \exp(tg)\ \mathbf{M}$$

est une courbe paramétrée lisse et

$$\frac{dM(t)}{dt}\Big|_{t=0}$$

définit un champ de vecteurs dans $\mathbb{R}^3$. Soit $\mathfrak{X}(\mathbb{R}^3)$ l'algèbre de Lie des champs de vecteurs dans $\mathbb{R}^3$. Nous avons la proposition suivante :

Proposition 5.2 – *Soit g appartenant à $\mathfrak{se}(3)$. Pour tout point M de $\mathbb{R}^3$, on note*

$$\varphi(g)(\mathbf{M}) = \frac{d}{dt}(\exp(tg)\mathbf{M})\Big|_{t=0}.$$

Alors, $\varphi(g) \in \mathfrak{X}(\mathbb{R}^3)$ et l'application φ définit un morphisme d'algèbres de Lie.

Remarques

– Le crochet de Lie de l'algèbre $\mathfrak{s}e(3)$ est donné par

$$\forall g, h \in \mathfrak{se}(3), \quad [g, h] = gh - hg.$$

– Le crochet de Lie de l'algèbre $\mathfrak{X}(\mathbb{R}^3)$ des champs de vecteurs de $\mathbb{R}^3$ est donné par

$$\forall \zeta, \xi \in \mathfrak{X}(\mathbb{R}^3),\ \forall x \in \mathbb{R}^3 \quad [\zeta, \xi](x) = d\zeta_x \xi(x) - d\xi_x \zeta(x).$$

Démonstration. Un morphisme d'algèbres de Lie est une application linéaire préservant le crochet de Lie. On a facilement la linéarité de φ car

$$\varphi(g)(\mathbf{M}) = \frac{d}{dt}(\exp(tg)\mathbf{M})\Big|_{t=0} = g\mathbf{M}.$$

De plus, $\forall g, h \in \mathfrak{se}(3),\ \forall M \in \mathbb{R}^3$,

$$\varphi([g, h])(\mathbf{M}) = [g, h]\mathbf{M} = (gh - hg)\mathbf{M},$$

et comme

$$d(\varphi(g))_M(\varphi(h)(\mathbf{M})) = g(\varphi(h)(\mathbf{M})) = gh\mathbf{M},$$

on a

$$[\varphi(g), \varphi(h)](\mathbf{M}) = d(\varphi(g))_M(\varphi(h)(\mathbf{M})) - d(\varphi(h))_M(\varphi(g)(\mathbf{M})) = \varphi([g,h])(\mathbf{M}).$$

Donc, le crochet de Lie est conservé et φ est un morphisme d'algèbres de Lie. □

Ainsi, à un mouvement infinitésimal de caméra, est associé un champ de vecteurs dans l'espace $\mathbb{R}^3$. Plus précisément, ce champ de vecteurs s'écrit $\frac{d}{dt}(\exp(tg)\mathbf{M})\Big|_{t=0}$ où les coordonnées des points M sont données dans le repère associé à la caméra.

5.2.2 Projection sur la sphère

Nous allons modifier un peu la construction précédente. Soit p la projection de $\mathbb{R}^3 \setminus \{0\}$ sur la sphère unité S^2

$$p(M) = \frac{M}{\|M\|}.$$

Par extension, on note

$$p(\mathbf{M}) = \begin{pmatrix} p(M) \\ 1 \end{pmatrix}.$$

Soit g appartenant à $\mathfrak{se}(3)$. La courbe paramétrée $\mathbf{M}(t) = \exp(tg)\mathbf{M}$ se projette en $p(M(t))$, et

$$\frac{d}{dt}\, p\left(\exp(tg)\mathbf{M}\right)\Big|_{t=0}$$

produit un vecteur sur S^2. Si r est une fonction strictement positive définie sur S^2, la fonction de S^2 dans S^2

$$\frac{d}{dt}\, p\left(\exp(tg)r(M)\mathbf{M}\right)\Big|_{t=0}$$

produit un champ de vecteurs sur S^2. Soit $\mathfrak{X}(S^2)$ l'algèbre de Lie des champs de vecteurs sur S^2.

Proposition 5.3 – *Soit g appartenant à $\mathfrak{se}(3)$ et r une fonction strictement positive définie sur S^2. Pour tout $M \in S^2$, on note*

$$\varphi_r(g)(\mathbf{M}) = \frac{d}{dt}\, p\left(\exp(tg)r(M)\mathbf{M}\right)\Big|_{t=0}.$$

Alors $\varphi_r(g) \in \mathfrak{X}(S^2)$ et l'application φ_r est un morphisme d'algèbres de Lie.

Démonstration. Soit $g = (\omega, v)$ appartenant à $\mathfrak{se}(3)$. $\forall M \in S^2$, on a

$$\varphi_r(g)(\mathbf{M}) = dp_M\left(r(M)\mathbf{M}\right) g\, r(M)\mathbf{M}$$

Comme

$$\forall M \in \mathbb{R}^3, \quad dp_M(\mathbf{M}) = \frac{1}{\|M\|} \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} - \frac{1}{\|M\|^3} \begin{pmatrix} MM^T & 0 \\ 0 & 0 \end{pmatrix},$$

on obtient, pour $M \in S^2$,

$$dp_M(r(M)\mathbf{M}) = \frac{1}{r(M)} \begin{pmatrix} I_3 - MM^T & 0 \\ 0 & 0 \end{pmatrix}$$

et

$$\varphi_r(g)(\mathbf{M}) = g\mathbf{M} - \langle v, M \rangle \begin{pmatrix} M \\ 0 \end{pmatrix}$$

où $\langle v, M \rangle = v_1 X + v_2 Y + v_3 Z$; ce qui prouve que la fonction φ_r est linéaire.

Montrons maintenant la conservation du crochet de Lie. Pour $g = (\omega, v)$ et $h = (\omega', v')$ appartenant à $\mathfrak{se}(3)$, $\forall M \in S^2$,

$$\varphi_r([g,h])(\mathbf{M}) = (gh - hg)\mathbf{M} - \langle [\omega]_\times v' - [\omega']_\times v, M \rangle \begin{pmatrix} M \\ 0 \end{pmatrix}.$$

Comme

$$d(\varphi_r(g))_M = g - \langle v, M \rangle \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} - \mathbf{M} \begin{pmatrix} v^T & 0 \end{pmatrix},$$

on a

$$d(\varphi_r(g))_M \varphi_r(h)(M) = \left(g - \langle v, M \rangle \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} - \mathbf{M} \begin{pmatrix} v^T & 0 \end{pmatrix} \right) \left(h\mathbf{M} - \langle v', M \rangle \begin{pmatrix} M \\ 0 \end{pmatrix} \right)$$

et le crochet de Lie des champs de vecteurs vaut
$d(\varphi_r(g))_M \varphi_r(h)(\mathbf{M}) - d(\varphi_r(h))_M \varphi_r(g)(\mathbf{M})$

$$= (gh - hg)\mathbf{M} - \mathbf{M} \left(\begin{pmatrix} v^T & 0 \end{pmatrix} h - \begin{pmatrix} v'^T & 0 \end{pmatrix} g \right) \mathbf{M}$$

$$= (gh - hg)\mathbf{M} - \mathbf{M} \begin{pmatrix} v^T [\omega']_\times - v'^T [\omega]_\times & 0 \end{pmatrix} \mathbf{M}$$

$$= (gh - hg)\mathbf{M} - \langle [\omega]_\times v' - [\omega']_\times v, M \rangle \begin{pmatrix} M \\ 0 \end{pmatrix}$$

$$= \varphi_r([g,h])(\mathbf{M}).$$

Le crochet de Lie est conservé par l'application φ_r ; φ_r étant linéaire, c'est un morphisme d'algèbres de Lie. □

On peut donc maintenant associer à un mouvement infinitésimal de caméra g un champ de vecteurs sur la sphère unité $\frac{d}{dt}\, p(\exp(tg) r(M)\mathbf{M})\Big|_{t=0}$ où $r(M)$ est la distance du point de l'espace $\mathbb{R}^3$ projeté en M sur S^2 au centre de la sphère, c'est-à-dire à la caméra.

5.2.3 Projection stéréographique

On utilise maintenant la composition de la projection p sur S^2 avec la projection sur le plan $\{Z = 0\}$. Soit q la projection stéréographique de $S^2 \setminus \{(0, 0, -1)\}$ sur $\{Z = 0\}$ définie par

$$q(M) = \frac{1}{1+Z} \begin{pmatrix} X \\ Y \\ 0 \end{pmatrix}.$$

Cette fonction envoie la demi-sphère supérieure $S^2 \cap \{Z \geq 0\}$ sur le disque $\{x^2 + y^2 \leq 1\}$ et la demi-sphère inférieure sur $\{x^2+y^2 \geq 1\}$. En utilisant p et q, on projette les courbes paramétrées $M(t)$ de $\mathbb{R}^3$ sur le plan $\{Z = 0\}$. Dans les chapitres précédents, nous avons considéré la projection directe des points de $\mathbb{R}^3$ sur le plan $\{Z = 1\}$. Cette projection est équivalente à la projection $q \circ p$, on explicitera par la suite le passage de l'une à l'autre.

Remarque – L'avantage principal de la projection q sur la projection sur $\{Z = 1\}$ est sa conformalité. Si par exemple, $M(t)$ est un cercle sur S^2, sa projection sur $\{Z = 0\}$ est encore un cercle. Un autre avantage de cette projection est de pouvoir prendre en compte la vision dans toutes les directions, aussi bien en avant qu'en arrière de la caméra [72].

Soit maintenant $\mathfrak{X}(\mathbb{R}^2)$ l'algèbre de Lie des champs de vecteurs de $\mathbb{R}^2$. On note $m = (x, y)$ les points du plan $\{Z = 0\}$. Si g appartient à $\mathfrak{se}(3)$ et r est une fonction de $\mathbb{R}^2$ dans $\mathbb{R}^*_+$, on considère la fonction de $\mathfrak{se}(3)$ dans $\mathfrak{X}(\mathbb{R}^2)$ définie par

$$\frac{d}{dt}\, q\left(p\left(\exp(tg) r(m) q^{-1}(m)\right)\right)\Big|_{t=0}.$$

À un mouvement de caméra infinitésimal est donc associé un champ de vecteurs sur $\mathbb{R}^2$. Sa construction sous-jacente est illustrée sur la figure (5.1). Soit une fonction strictement positive r définie sur $\mathbb{R}^2$, représentant la distance à la caméra des points projetés par $q \circ p$ sur $\{Z = 0\}$ et soit un mouvement infinitésimal g de $\mathfrak{se}(3)$. Le champ de vecteurs dans $\mathbb{R}^2$ est construit comme suit. Prenons un point m de $\mathbb{R}^2$. Il correspond au point $q^{-1}(m)$ de S^2 et au point $r(m)q^{-1}(m)$ de $\mathbb{R}^3$. Par l'action du mouvement infinitésimal g, le point de $\mathbb{R}^3$ a pour trajectoire la courbe paramétrée $\exp(tg)\, r(m)\, q^{-1}(m)$, que l'on projette sur S^2 en $p\left(\exp(tg)\, r(m)\, q^{-1}(m)\right)$ et finalement sur le plan $\{Z = 0\}$ en $m(t) = q\left(p\left(\exp(tg)\, r(m)\, q^{-1}(m)\right)\right)$. L'ensemble des vecteurs vitesses $\frac{d}{dt}\, m(t)\big|_{t=0}$ associés aux points de $\mathbb{R}^2$ produit sur le plan $\{Z = 0\}$ le champ de vecteurs appelé flot optique.

5.2.4 Flot optique

Proposition 5.4 – *Soit $g = (\omega, v)$ appartenant à $\mathfrak{se}(3)$, r une fonction strictement positive définie sur $\mathbb{R}^2$ et*

$$u(m) = \frac{d}{dt}\, q \circ p\left(\exp(tg)\, r(m)\, q^{-1}(m)\right)\Big|_{t=0}.$$

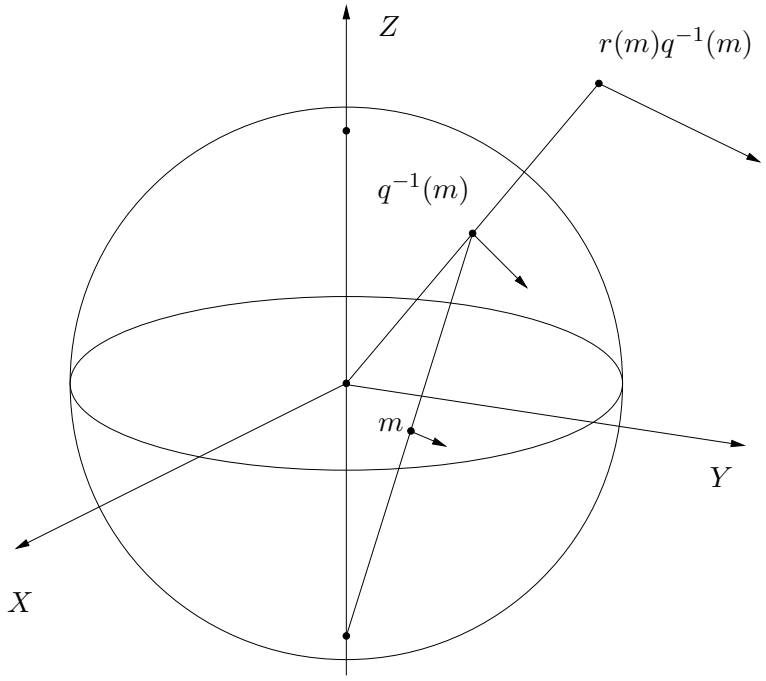

FIGURE 5.1: Construction du champ de vecteurs sur le plan $\{Z = 0\}$ associé au déplacement infinitésimal g dans $\mathbb{R}^3$.

Alors, en utilisant la notation complexe dans $\mathbb{R}^2$ *(on note* $m = (x, y)$ *par* $m = x + iy$*),*

$$\begin{aligned} u(m) &= \frac{\omega_2 + i\omega_1}{2}\, m^2 + i\omega_3\, m + \frac{\omega_2 - i\omega_1}{2} + \frac{1}{r(m)}\left(\frac{-v_1 + iv_2}{2}\, m^2 - v_3\, m + \frac{v_1 + iv_2}{2}\right) \\ &= \phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m). \end{aligned}$$

On appelle $\phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m)$ *le flot optique associé au mouvement* $g = (\omega, v)$ *et à la surface* r*.*

Démonstration. Soit $M = (X, Y, Z)$ un point de $\mathbb{R}^3$ et $m = (x, y)$ sa projection sur $\{Z = 0\}$. On note $\|M\| = r(m)$. Alors

$$m = q \circ p(M) = \begin{pmatrix} \dfrac{X}{r(m) + Z} \\ \\ \dfrac{Y}{r(m) + Z} \end{pmatrix}$$

ce qui équivaut à $M = q^{-1}(m)\, r(m)$. Considérons la trajectoire $\mathbf{M}(t) = \exp(tg)\,\mathbf{M}$ dans $\mathbb{R}^3$ et

sa projection sur $\{Z = 0\}$, $m(t) = q \circ p(M(t))$. Notons

$$\exp t \begin{pmatrix} [\omega]_\times & u \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} R(t) & T(t) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_1(t) & T_1(t) \\ R_2(t) & T_2(t) \\ R_3(t) & T_3(t) \\ 0 & 1 \end{pmatrix}.$$

Avec cette notation,

$$q \circ p(M(t)) = \frac{1}{R_3(t)M + T_3(t) + \|M(t)\|} \begin{pmatrix} R_1(t)M + T_1(t) \\ R_2(t)M + T_2(t) \end{pmatrix}.$$

Comme

$$\begin{cases} R(0) = I_3 & T(0) = 0 \\ \\ \dfrac{R(t)}{dt}\Big|_{t=0} = [\omega]_\times & \dfrac{T(t)}{dt}\Big|_{t=0} = v, \end{cases}$$

on a

$$\frac{d}{dt}\left(\frac{R_1(t)M + T_1(t)}{R_3(t)M + T_3(t) + \|M(t)\|}\right)\Bigg|_{t=0}$$

$$= \frac{-\omega_3 Y + \omega_2 Z + v_1}{Z + \|M\|} - \frac{X\left(-\omega_2 X + \omega_1 Y + v_3 + \frac{d\|M(t)\|}{dt}\Big|_{t=0}\right)}{(Z + \|M\|)^2}.$$

Or,

$$\frac{d\|M(t)\|}{dt}\Big|_{t=0} = \frac{v_1 X + v_2 Y + v_3 Z}{\|M\|} \quad \text{et} \quad \|M\| = r(m)$$

donc

$$\frac{d}{dt}\left(\frac{R_1(t)M + T_1(t)}{R_3(t)M + T_3(t) + \|M(t)\|}\right)\Bigg|_{t=0} =$$

$$\frac{\omega_2}{2}(x^2 - y^2 + 1) - \omega_1 xy - \omega_3 y + \frac{v_1 - v_3 x}{Z + r(m)} - \frac{1}{r(m)}(v_1 x^2 + v_2 xy) - \frac{v_3 x(1 - x^2 - y^2)}{2r(m)}.$$

Comme

$$Z = \frac{1 - x^2 - y^2}{1 + x^2 + y^2}\, r(m),$$

l'expression devient

$$\frac{d}{dt}\left(\frac{R_1(t)M + T_1(t)}{R_3(t)M + T_3(t) + \|M(t)\|}\right)\Bigg|_{t=0} =$$

$$-\omega_1 xy + \frac{\omega_2}{2}(x^2 - y^2 + 1) - \omega_3 y - \frac{1}{r(m)}\left(\frac{v_1}{2}(x^2 - y^2 - 1) + v_2 xy - v_3 x\right).$$

De la même façon, on obtient

$$\frac{d}{dt}\left(\frac{R_2(t)M + T_2(t)}{R_3(t)M + T_3(t) + \|M(t)\|}\right)\Big|_{t=0} =$$

$$\frac{\omega_1}{2}(x^2 - y^2 - 1) + \omega_2 xy + \omega_3 x - \frac{1}{r(m)}\left(v_1 xy - \frac{v_2}{2}(x^2 - y^2 + 1) + v_3 y\right).$$

ce qui termine la preuve. □

Remarques

- L'expression du flot optique sur le plan $\{Z = 0\}$ est complexe quadratique et sépare le flot en deux composantes indépendantes, l'une due à la rotation infinitésimale ω, l'autre due à la translation infinitésimale v.
- Le cercle unité $\mathcal{H}$ de $\{Z = 0\}$ joue un rôle particulier dans ces constructions. En effet, si la caméra, placée à l'origine, peut seulement voir dans la direction de $(0, 0, 1)$, le plan $\{Z = 0\}$ est la limite du visible (à moins d'être une mouche). La projection de $\{Z = 0\}$ sur S^2 est le cercle $\mathcal{H}$. C'est pourquoi on l'appellera l'horizon.

Exemple – Étudions les flots optiques obtenus pour des choix particuliers de ω, v et r. Si $r \equiv 1$, $\omega = (0, 1, 0)$ et $v = (0, 0, 0)$, alors

$$\phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = \frac{m^2}{2} + \frac{1}{2}.$$

De la même façon, si $\omega = (0, 0, 0)$ et $v = (1, 0, 0)$

$$\phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = -\frac{m^2}{2} + \frac{1}{2}.$$

Ces deux flots optiques sont représentés sur la figure (5.2).

Travailler avec les flots optiques $\phi_\omega + \frac{1}{r}\,\psi_v$ nécessite quelques définitions supplémentaires. D'abord, pour une fonction r constante, si $\omega = (0, 0, 0)$, on appellera le flot $\phi_\omega + \frac{1}{r}\,\psi_v$ une translation et si $v = (0, 0, 0)$, une rotation. On peut facilement vérifier que les translations et les rotations ont des pôles, c'est-à-dire des points où le flot est nul. Ceci est très clair quand les champs de vecteurs sont représentés sur S^2, voir [49, 7]. Une translation et une rotation ont toujours exactement deux pôles, à l'exception des rotations d'axe Z et des translations de direction Z. Dans ces deux cas particuliers, l'un des pôles est envoyé à l'infini et l'autre est à l'origine du plan $\{Z = 0\}$. Dans le cas d'une rotation quelconque, les pôles sur S^2 sont les intersections de S^2 avec la droite passant par l'origine et dirigée suivant ω. Pour une translation, ce sont les intersections de S^2 avec la droite passant par l'origine et dirigée suivant v. Les pôles d'une translation ou d'une rotation sont donc opposés sur S^2 et inverses par rapport au cercle

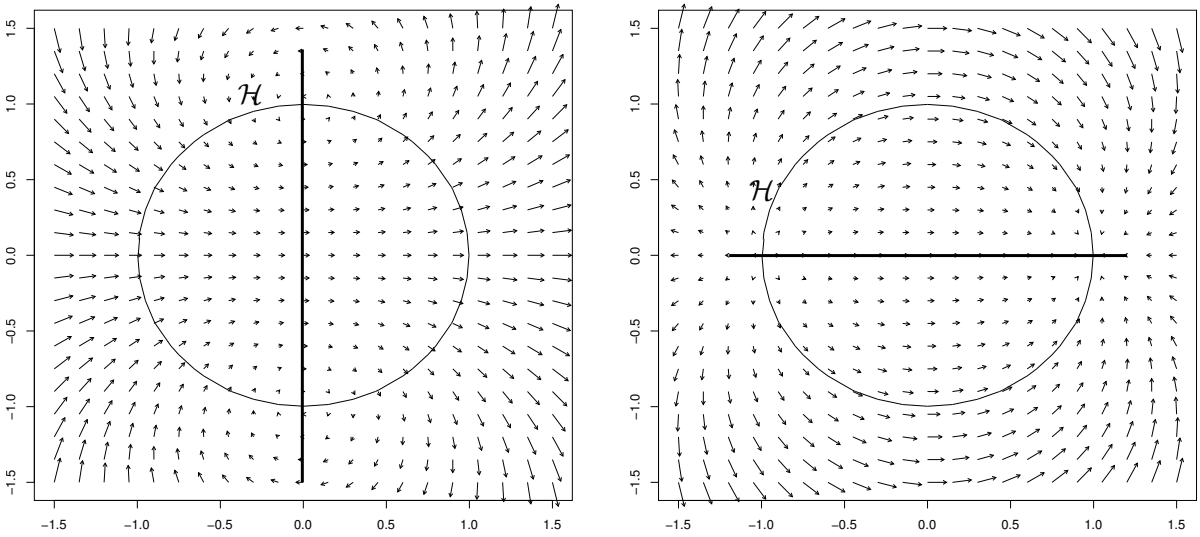

FIGURE 5.2: *Flots optiques sur* $\{Z = 0\}$*, à gauche pour* $\omega = (0, 1, 0)$*,* $v = (0, 0, 0)$ *et* $r \equiv 1$ *et à droite pour* $\omega = (0, 0, 0)$*,* $v = (1, 0, 0)$ *et* $r \equiv 1$*. Dans les deux cas, les trajectoires sont des cercles. L'équateur de la rotation à gauche et celui de la translation à droite sont représentés en gras.*

$\mathcal{H}$ de $\mathbb{R}^2$. Ceci permet de définir l'équateur d'une translation et d'une rotation comme la droite passant par les pôles (représenté en gras sur la figure (5.2)). Dans la suite, on notera ω^+, ω^- les pôles d'une rotation définie par le vecteur ω et v^+, v^- les pôles d'une translation définie par v. Le pôle ω^+ désigne le pôle autour duquel la rotation est directe et v^+ le pôle attractif de la translation.

On appellera grands cercles la projection sur $\{Z = 0\}$ de cercles géodésiques de S^2. Les équateurs des translations et des rotations par exemple sont des grands cercles.

Enfin, on dira que deux cercles du plan sont orthogonaux s'ils s'intersectent en un angle droit. Comme la projection q est conforme, des cercles orthogonaux dans S^2 produisent des cercles orthogonaux dans $\{Z = 0\}$.

Remarque – Soit r une fonction strictement positive définie sur $\mathbb{R}^2$ et g appartenant à $\mathfrak{se}(3)$. Pour tout $m \in \mathbb{R}^2$, on note

$$\varphi_r(g)(m) = \frac{d}{dt}\, q\left(p\left(\exp(tg)\, r(m)\, q^{-1}(m)\right)\right)\Big|_{t=0}.$$

Alors $\varphi_r(g)$ appartient à $\mathfrak{X}(\mathbb{R}^2)$ mais la fonction φ_r n'est pas un morphisme d'algèbres de Lie car le crochet de Lie n'est pas conservé par φ_r. Prenons par exemple $r \equiv 1$, $g = (\omega, v)$ et $h = (\omega', v')$. En notation complexe, on a

$$[\varphi(g), \varphi(h)](m) = \varphi(gh - hg)(m) - im(v_1 v_2' - v_2 v_1') - \frac{1+m^2}{2}(v_1 v_3' - v_3 v_1') + i\,\frac{1-m^2}{2}(v_2 v_3' - v_3 v_2'),$$

donc le crochet de Lie n'est pas conservé par les translations.

5.2.5 Projections stéréographique et sténopé

Soit un point M de $\mathbb{R}^3$ projeté en m sur le plan $\{Z = 0\}$, suivant la projection $q \circ p$, et en m' sur le plan $\{Z = 1\}$. Le point m est indifféremment la projection du point M ou du point m', comme illustré sur la figure (5.3). Ainsi, $m = q \circ p(M) = q \circ p(m')$.

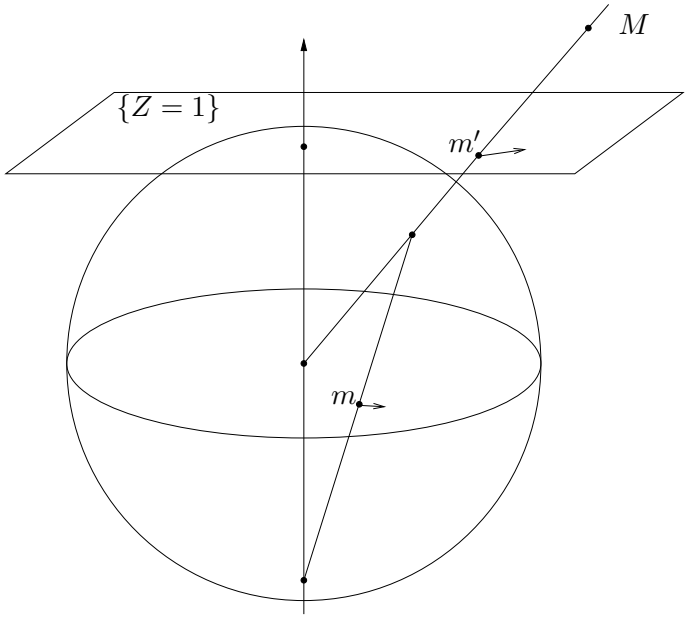

FIGURE 5.3: *Projection sur la sphère suivie de la projection stéréographique sur* $\{Z = 0\}$ *d'un point* M *et projection sténopé du même point* M *sur* $\{Z = 1\}$.

Il est équivalent de considérer la projection stéréographique ou sténopé car l'application $q \circ p$ de $\{Z = 1\}$ sur le disque unité du plan $\{Z = 0\}$ est inversible. En effet, en notant $m = (x, y)$, $m' = (x', y')$ et $\mathbf{m}' = (x', y', 1)$, on a

$$m = q \circ p(\mathbf{m}') = \begin{pmatrix} \dfrac{x'}{1 + \|\mathbf{m}'\|} \\ \\ \dfrac{y'}{1 + \|\mathbf{m}'\|} \end{pmatrix} \quad \text{avec} \quad \|\mathbf{m}'\| = \sqrt{1 + x'^2 + y'^2}.$$

Or,

$$d(q \circ p)_{m'} = \begin{pmatrix} \dfrac{1 + \|\mathbf{m}'\| + y'^2}{\|\mathbf{m}'\| \ (1 + \|\mathbf{m}'\|)^2} & -\dfrac{x'y'}{\|\mathbf{m}'\| \ (1 + \|\mathbf{m}'\|)^2} \\ \\ -\dfrac{x'y'}{\|\mathbf{m}'\| \ (1 + \|\mathbf{m}'\|)^2} & \dfrac{1 + \|\mathbf{m}'\| + x'^2}{\|\mathbf{m}'\| \ (1 + \|\mathbf{m}'\|)^2} \end{pmatrix},$$

le déterminant de la matrice jacobienne $d(q \circ p)_{m'}$ est non nul, donc l'application $q \circ p$ est inversible. Le flot optique u obtenu sur le disque unité du plan $\{Z = 0\}$ est lié au flot optique u' sur le plan $\{Z = 1\}$ par

$$u(m) = u(q \circ p(m')) = d(q \circ p)_{m'}\, u'(m').$$

5.3 Injectivité du flot optique

On s'interroge sur la possibilité qu'un flot optique donné $\phi_\omega + \frac{1}{r}\,\psi_v$ puisse résulter de différents mouvements et profondeurs. Il faut bien sûr garder à l'esprit un cas trivial : si $\lambda > 0$ alors

$$\phi_\omega + \frac{1}{r}\,\psi_v \text{ et } \phi_\omega + \frac{1}{\lambda r}\,\psi_{\lambda v}$$

produiront le même flot optique. Ceci étant, nous allons montrer le résultat suivant.

Théorème 5.1 – *Soient $\omega, \omega', v, v' \in \mathbb{R}^3$ et r, s deux fonctions strictement positives définies sur $\mathbb{R}^2$. Si*

$$\forall m \in \mathbb{R}^2,\ \ \phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = \phi_{\omega'}(m) + \frac{1}{s(m)}\,\psi_{v'}(m)$$

alors, $\omega = \omega'$ et il existe $\lambda > 0$ tel que $v = \lambda v'$ et $r = \lambda s$.

Pour prouver ce théorème, nous allons d'abord montrer le lemme suivant.

Lemme 5.1 – *Soient $\omega, v, v' \in \mathbb{R}^3$ et r, s deux fonctions strictement positives définies sur $\mathbb{R}^2$. Si*

$$\forall m \in \mathbb{R}^2,\ \ \phi_\omega(m) = \frac{1}{r(m)}\,\psi_v(m) + \frac{1}{s(m)}\,\psi_{v'}(m) \tag{5.1}$$

alors $\omega = 0$ et il existe $\lambda > 0$ tel que $r = \lambda s$ et $v = -\lambda v'$.

Démonstration. A : Supposons d'abord que ω, v et v' sont non nuls. D'après l'équation (5.1), en chaque point m de $\mathbb{R}^2$, le vecteur $\phi_\omega(m)$ est décomposé sur $\psi_v(m)$ et $\psi_{v'}(m)$ avec des coefficients positifs. Ceci implique

$$\operatorname{sgn}\left(\det(\psi_v(m), \phi_\omega(m))\right) = \operatorname{sgn}\left(\det(\phi_\omega(m), \psi_{v'}(m))\right)$$

car le vecteur $\phi_\omega(m)$ est situé à l'intérieur du secteur angulaire formé par les vecteurs $\psi_v(m)$ et $\psi_{v'}(m)$, comme illustré sur la figure (5.4).
B : On considère l'ensemble

$$\mathcal{E}_{\omega v} = \{m \in \mathbb{R}^2 \text{ t. q. } \det(\psi_v(m), \phi_\omega(m)) = 0\}.$$

C'est l'ensemble des points de tangence des champs de vecteurs ψ_v et ϕ_ω. Cet ensemble contient aussi les quatre pôles ω^+, ω^-, v^+ et v^- (sauf s'ils sont à l'infini), comme le montrent les exemples

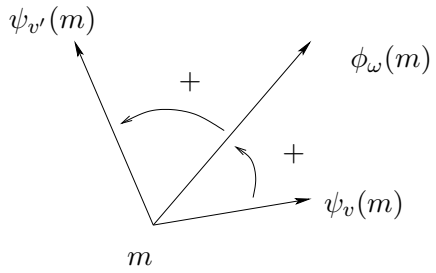

FIGURE 5.4: *Si* $\phi_\omega(m) = \frac{1}{r(m)}\,\psi_v(m) + \frac{1}{s(m)}\,\psi_{v'}(m)$, *avec* $s(m) > 0$ *et* $r(m) > 0$, *alors* $\det(\psi_v(m), \phi_\omega(m))$ *et* $\det(\phi_\omega(m), \psi_{v'}(m))$ *doivent avoir le même signe.*

présentés sur la figure (5.5). Les points de $\mathcal{E}_{\omega v}$ satisfont l'équation suivante

$$\left(\tfrac{x^2+y^2+1}{2}\right)^2 (\omega_1 v_1 + \omega_2 v_2) - x^2\omega_1 v_1 - y^2\omega_2 v_2 - xy\,(\omega_1 v_2 + v_1\omega_2)$$

$$+\tfrac{x^2+y^2-1}{2}\,(x(\omega_1 v_3 + v_1\omega_3) + y(\omega_2 v_3 + \omega_3 v_2)) + (x^2+y^2)\,\omega_3 v_3 = 0.$$

Le déterminant $\det(\psi_v, \phi_\omega)$ est de signe constant dans chaque domaine limité par $\mathcal{E}_{\omega v}$, par

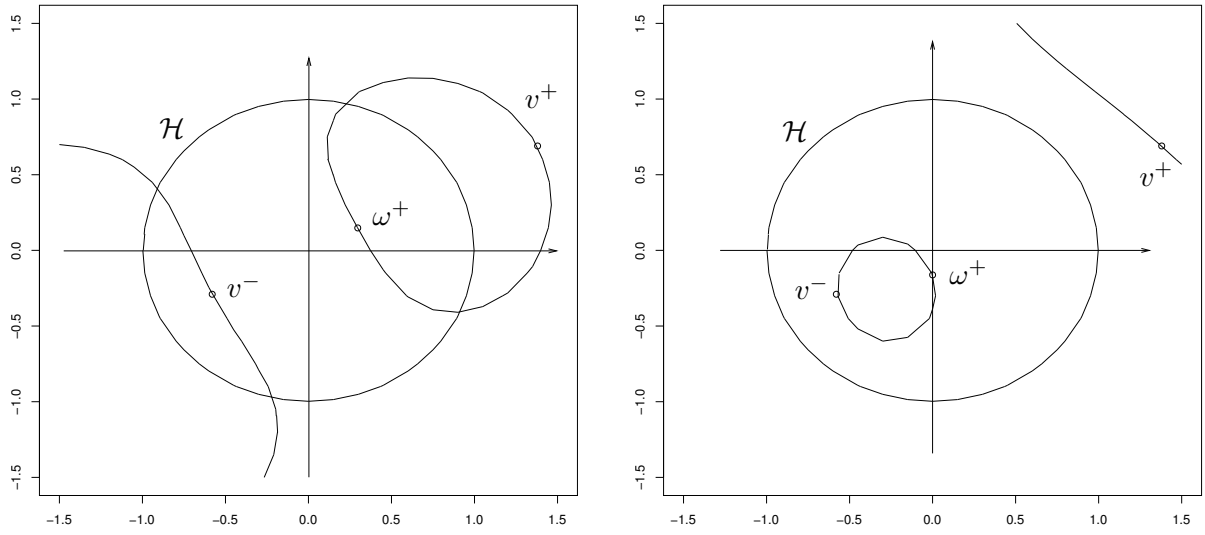

FIGURE 5.5: *Deux exemples d'ensembles* $\mathcal{E}_{\omega v}$ *; à gauche, pour* $\omega = (2,1,3)$ *et* $v = (2,1,-1)$ *et à droite, pour* $\omega = (0,-1,3)$ *et* $v = (2,1,-1)$.

continuité de l'application $\det(\psi_v, \phi_\omega)$. Montrons que $\mathcal{E}_{\omega v}$ sépare le plan $\{Z = 0\}$ en régions où le déterminant a des signes opposés. Il suffit de le montrer dans le cas où tous les pôles sont situés sur le cercle unité $\mathcal{H}$. Par conjugaison, cette propriété sera vérifiée pour toute configuration des pôles. Dans le cas où les pôles sont situés sur $\mathcal{H}$, on a $v_3 = \omega_3 = 0$, donc $\omega_1 + i\omega_2 = \|\omega\| e^{i\alpha}$ et

$v_1 + iv_2 = \|v\|e^{i\nu}$. Alors, en notation complexe,

$$\begin{cases} \phi_\omega(m) = \dfrac{i}{2}\left(m^2 e^{-i\alpha} - e^{i\alpha}\right)\|\omega\| \\ \\ \psi_v(m) = \dfrac{-1}{2}\left(m^2 e^{-i\nu} - e^{i\nu}\right)\|v\|. \end{cases}$$

Calculons maintenant l'angle entre ϕ_ω et ψ_v, qui nous donnera le signe de $\det(\psi_v, \phi_\omega)$. Si m est un point de $\mathcal{H}$, on peut écrire $m = e^{i\delta}$, alors

$$\arg \frac{\phi_\omega(e^{i\delta})}{\psi_v(e^{i\delta})} = \arg \frac{i(e^{2i\delta}e^{-i\alpha} - e^{i\alpha})\|\omega\|}{-(e^{2i\delta}e^{-i\nu} - e^{i\nu})\|v\|} = \arg\left(\frac{\sin(\delta-\alpha)\,\|\omega\|}{i\sin(\delta-\nu)\,\|v\|}\right)$$

$$= \begin{cases} \dfrac{-\pi}{2} & \text{si} \quad \delta \in (]\nu, \nu+\pi[\cap]\alpha, \alpha+\pi[) \cup \\ & \qquad (]-\pi+\nu, \nu[\cap]-\pi+\alpha, \alpha[) \\ \\ \dfrac{\pi}{2} & \text{si} \quad \delta \in (]\nu, \nu+\pi[\cap]-\pi+\alpha, \alpha[) \cup \\ & \qquad (]-\pi+\nu, \nu[\cap]\alpha, \alpha+\pi[) \\ \\ 0 & \text{si} \quad \delta \in \{\alpha, \nu, \alpha+\pi, \nu+\pi\} \end{cases}$$

Ainsi, sur le cercle $\mathcal{H}$, le déterminant $\det(\psi_v, \phi_\omega)$ a même signe pour les points situés entre ω^+ et v^+, ω^- et v^-, il est nul aux pôles et a le signe opposé ailleurs, comme illustré sur l'exemple de la figure (5.6). On peut en déduire que $\mathcal{E}_{\omega v}$ sépare $\mathbb{R}^2$ en régions où le déterminant a des signes opposés.

Considérons maintenant $\mathcal{E}_{\omega v'}$. Si $\mathcal{E}_{\omega v} \neq \mathcal{E}_{\omega v'}$, alors il existe des points où $\det(\psi_v, \phi_\omega)$ et $\det(\phi_\omega, \psi_{v'})$ ont des signes différents, ce qui est impossible, d'après (A). On a donc $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$.

C : Montrons maintenant que $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$ implique $\{v^+, v^-\} = \{v'^+, v'^-\}$. Considérons le pôle ω^+. Comme $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$, les tangentes en ω^+ à $\mathcal{E}_{\omega v}$ et $\mathcal{E}_{\omega v'}$ doivent être parallèles, ce qui équivaut à

$$\det \begin{pmatrix} \dfrac{\partial}{\partial x}\det(\psi_v(m), \phi_\omega(m)) & \dfrac{\partial}{\partial x}\det(\phi_\omega(m), \psi_{v'}(m)) \\ \\ \dfrac{\partial}{\partial y}\det(\psi_v(m), \phi_\omega(m)) & \dfrac{\partial}{\partial y}\det(\phi_\omega(m), \psi_{v'}(m)) \end{pmatrix}\Bigg|_{m=\omega^+} = 0$$

$$\Leftrightarrow \frac{\|\omega\|^3}{(\omega_3 + \|\omega\|)^2}\det(\omega, v, v') = 0.$$

Examinons d'abord le cas où $\omega_3 + \|\omega\| = 0$. Ceci n'est possible que si $\omega_1 = \omega_2 = 0$ et $\omega_3 < 0$. Dans ce cas, le pôle ω^+ est envoyé à l'infini et on doit alors considérer le pôle ω^- (qui est alors à l'origine du plan $\{Z = 0\}$).

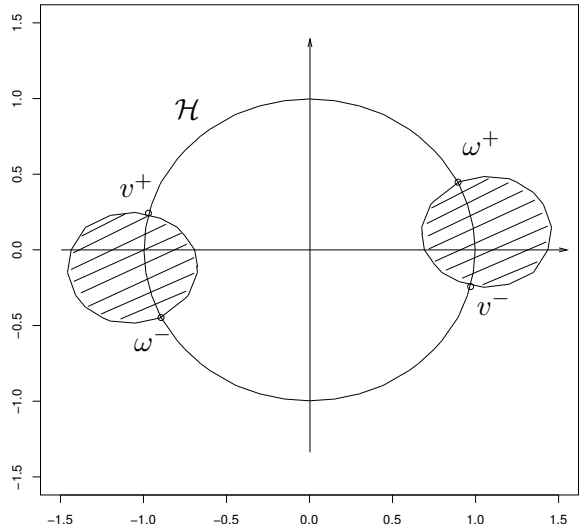

FIGURE 5.6: *Pour $\omega = (2, 1, 0)$ et $v = (-4, 1, 0)$, les pôles sont placés sur $\mathcal{H}$. Le déterminant* $\det(\psi_v, \phi_\omega)$ *est strictement positif dans les ensembles hachurés définis par $\mathcal{E}_{\omega v}$, nul sur $\mathcal{E}_{\omega v}$ et négatif ailleurs.*

Maintenant, si $\omega_3 + \|\omega\| \neq 0$, alors $\det(\omega, v, v') = 0$, ce qui implique que les vecteurs ω, v et v' sont liés, c'est-à-dire qu'il existe $(\alpha, \nu, \mu) \neq (0, 0, 0)$ tels que

$$\alpha\omega + \nu v + \mu v' = 0.$$

Montrons que $\{v^-, v^+\} = \{v'^-, v'^+\}$.

- Si $\alpha = 0$ alors $\nu \neq 0$ et $\mu \neq 0$ car v et v' sont non nuls. Donc, v et v' sont colinéaires, ce qui implique que les pôles de v et v' sont confondus.
- Si $\mu = 0$ alors ω et v sont colinéaires et ont donc mêmes pôles. On peut vérifier dans ce cas que $\mathcal{E}_{\omega v}$ n'est formé que des pôles de ω et v car les vecteurs $\psi_v(m)$ et $\phi_\omega(m)$ sont orthogonaux pour tout $m \in \mathbb{R}^2 \setminus \{v^-, v^+\}$. Comme $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$, $\mathcal{E}_{\omega v'}$ est formé de deux points qui sont aussi ses pôles. En conséquence, v et v' ont mêmes pôles. On utilise le même raisonnement pour le cas $\nu = 0$.
- Si $\alpha \neq 0$, $\mu \neq 0$ et $\nu \neq 0$, on peut vérifier que les pôles de la rotation et des deux translations appartiennent à un même grand cercle. Supposons que ce grand cercle soit $\mathcal{H}$. Comme $\mathcal{E}_{\omega v} \cap \mathcal{H} = \{\omega^+, \omega^-, v^+, v^-\}$ et $\mathcal{E}_{\omega v'} \cap \mathcal{H} = \{\omega^+, \omega^-, v'^+, v'^-\}$ (d'après B), $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$ implique $\{v^+, v^-\} = \{v'^+, v'^-\}$. Par conjugaison, ceci est vérifié pour tout grand cercle contenant les six pôles.

D : Les vecteurs v et v' sont colinéaires donc le flot optique ϕ_ω est égal au flot optique généré par une translation pondérée. Mais le flot généré par une rotation non nulle ne peut être proportionnel à celui généré par une translation, donc $\omega = 0$. D'où,

$$\forall m \in \mathbb{R}^2, \quad \frac{1}{r(m)}\,\psi_v(m) + \frac{1}{s(m)}\,\psi_{v'}(m) = 0.$$

Comme v et v' sont colinéaires et r et s sont positives, il existe λ positif tel que

$$v = -\lambda v' \text{ et } r = \lambda s.$$

Le lemme est prouvé. □

Nous pouvons maintenant démontrer le théorème 1.

Démonstration. Si $\forall m \in \mathbb{R}^2$,

$$\phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = \phi_{\omega'}(m) + \frac{1}{s(m)}\,\psi_{v'}(m),$$

alors

$$\phi_{\omega-\omega'}(m) = \frac{1}{r(m)}\,\psi_{-v}(m) + \frac{1}{s(m)}\,\psi_{v'}(m).$$

En appliquant le lemme 1, on obtient $\omega - \omega' = 0$, $v = \lambda v'$ et $r = \lambda s$. □

Nous avons aussi un résultat plus fort.

Théorème 5.2 – *Soient $\omega, \omega', v, v' \in \mathbb{R}^3$, D^2 le disque unité ouvert de $\mathbb{R}^2$ et r, s deux fonctions strictement positives définies sur D^2. Si*

$$\forall m \in D^2,\ \ \phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = \phi_{\omega'}(m) + \frac{1}{s(m)}\,\psi_{v'}(m)$$

il existe $\lambda > 0$ tel que $\omega = \omega'$, $v = \lambda v'$ et $r = \lambda s$.

Ce résultat est obtenu grâce au fait que les pôles sont inverses par rapport au cercle $\mathcal{H}$. Nous avons le lemme suivant.

Lemme 5.2 – *Soient $\omega, v, v' \in \mathbb{R}^3$ et r, s deux fonctions strictement positives définies sur D^2. Si*

$$\forall m \in D^2,\ \ \phi_\omega(m) = \frac{1}{r(m)}\,\psi_v(m) + \frac{1}{s(m)}\,\psi_{v'}(m)$$

alors $\omega = 0$ et il existe $\lambda > 0$ tel que $r = \lambda s$ et $v = -\lambda v'$.

Démonstration. Comme les pôles sont inverses par rapport à $\mathcal{H}$, l'un des pôles de ω est toujours à l'intérieur du disque unité fermé. Si ω^- ou ω^+ est dans le disque ouvert, on peut appliquer la preuve du lemme 1 (en montrant que $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$ puis en calculant les tangentes à $\mathcal{E}_{\omega v}$ et $\mathcal{E}_{\omega v'}$ en ω^+ ou ω^-).

Mais on peut aussi avoir les deux pôles de ω sur les bords de D^2 qui est le cercle $\mathcal{H}$. Dans ce cas, un pôle de v, disons v^+, se trouve dans le cercle unité fermé. Alors, si $\mathcal{E}_{\omega v} \neq \mathcal{E}_{\omega v'}$, on peut trouver un disque ouvert $B(v^+, \varepsilon)$, avec $\varepsilon > 0$, tel que $B(v^+, \varepsilon) \cap D^2 \neq \emptyset$, $\det(\phi_\omega, \psi_{v'})$ ait un signe constant et $\det(\psi_v, \phi_\omega)$ ait des signes différents dans $B(v^+, \varepsilon)$. En conséquence, $\mathcal{E}_{\omega v} = \mathcal{E}_{\omega v'}$ ce qui mène au résultat. □

5.4 Flots optiques ambigus

5.4.1 Domaine d'observation ambigu

Les résultats précédents suggèrent qu'on puisse perdre l'injectivité dans un domaine spécifique d'observation du flot optique.

Théorème 5.3 – *Soit $g = (\omega, v)$ et $h = (\omega', v')$ appartenant à $\mathfrak{se}(3)$. Les flots optiques générés par ces deux mouvements peuvent être égaux dans une région U du plan $\{Z = 0\}$ si et seulement si*

$$U \subset (\mathcal{E}^+_{\omega-\omega',v} \cap \mathcal{E}^+_{\omega-\omega',v'}) \cup (\mathcal{E}^-_{\omega-\omega',v} \cap \mathcal{E}^-_{\omega-\omega',v'}),$$

où

$$\mathcal{E}^+_{\omega,v} = \{m \in \mathbb{R}^2 \text{ tels que } \det(\psi_v, \phi_\omega) \geq 0\}$$

et

$$\mathcal{E}^-_{\omega,v} = \{m \in \mathbb{R}^2 \text{ tels que } \det(\psi_v, \phi_\omega) < 0\}.$$

Démonstration. Si dans U,

$$\phi_\omega + \frac{1}{r}\psi_v = \phi_{\omega'} + \frac{1}{s}\psi_{v'}$$

alors pour $m \in U$, on peut décomposer $\phi_{\omega-\omega'}(m)$ sur $-\psi_v(m)$ et $\psi_{v'}(m)$ avec des coefficients positifs $\frac{1}{r(m)}$ et $\frac{1}{s(m)}$. Par conséquent, pour $m \in U$,

$$\operatorname{sgn}\left(\det(-\psi_v(m), \phi_{\omega-\omega'}(m))\right) = \operatorname{sgn}\left(\det(\phi_{\omega-\omega'}(m), \psi_{v'}(m))\right),$$

c'est-à-dire

$$\operatorname{sgn}\left(\det(\psi_v(m), \phi_{\omega-\omega'}(m))\right) = \operatorname{sgn}\left(\det(\psi_{v'}(m), \phi_{\omega-\omega'}(m))\right),$$

ce qui implique $U \subset (\mathcal{E}^+_{\omega-\omega',v} \cap \mathcal{E}^+_{\omega-\omega',v'}) \cup (\mathcal{E}^-_{\omega-\omega',v} \cap \mathcal{E}^-_{\omega-\omega',v'})$. □

Un exemple de domaine d'observation ambigu du flot optique associé à deux mouvements infinitésimaux est présenté sur la figure (5.7).

5.4.2 Surfaces filmées ambiguës

Si deux flots optiques générés par deux mouvements donnés sont identiques dans un domaine U du plan $\{Z = 0\}$, il est facile de calculer les surfaces r et s filmées.

Exemple – Prenons $\omega = (0, 1, 1)$, $\omega' = (0, 0, 1)$, $v = (-1, -1, 0)$ et $v' = (1, -1, 0)$. Le domaine du plan $\{Z = 0\}$ où les flots optiques sont susceptibles d'être égaux est représenté sur la figure (5.7). On a alors

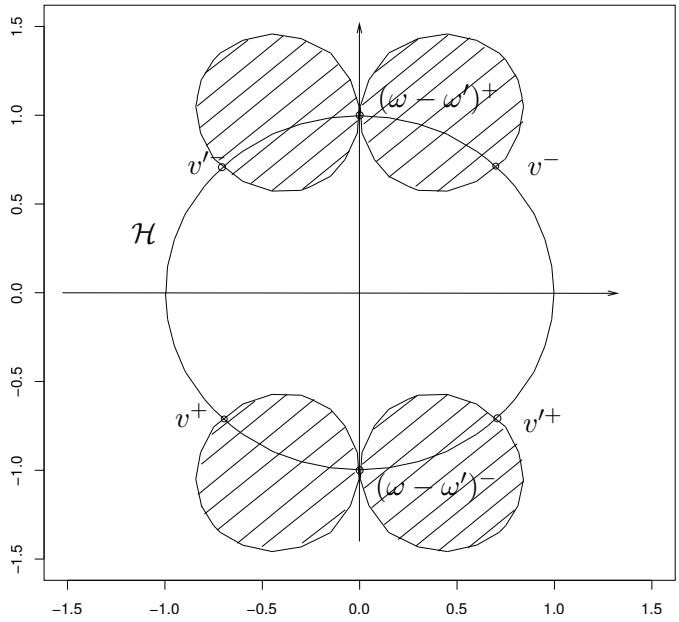

FIGURE 5.7: *Le domaine hachuré est le domaine où les flots optiques générés par* $(\omega, v) = ((0,1,1),(-1,-1,0))$ *et* $(\omega', v') = ((0,0,1),(1,-1,0))$ *ne peuvent pas être égaux. Sur le reste du domaine, l'observation peut être ambiguë.*

$$\phi_{\omega-\omega'}(m) = \begin{pmatrix} \dfrac{x^2-y^2+1}{2} \\ \\ xy \end{pmatrix},$$

$$\psi_{-v}(m) = \begin{pmatrix} \dfrac{-x^2+y^2+1}{2} - xy \\ \\ -xy + \dfrac{x^2-y^2+1}{2} \end{pmatrix},$$

$$\psi_{v'}(m) = \begin{pmatrix} \dfrac{-x^2+y^2+1}{2} + xy \\ \\ -xy - \dfrac{x^2-y^2+1}{2} \end{pmatrix},$$

et on cherche $\begin{pmatrix} r(x,y) \\ s(x,y) \end{pmatrix}$ tels que

$$\begin{pmatrix} \frac{-x^2+y^2+1}{2} - xy & \frac{-x^2+y^2+1}{2} + xy \\ -xy + \frac{x^2-y^2+1}{2} & -xy - \frac{x^2-y^2+1}{2} \end{pmatrix} \begin{pmatrix} \frac{1}{r(x,y)} \\ \frac{1}{s(x,y)} \end{pmatrix} = \begin{pmatrix} \frac{x^2-y^2+1}{2} \\ xy \end{pmatrix}.$$

On obtient ainsi

$$\begin{cases} r(x,y) = \dfrac{2((x^2+y^2)^2-1)}{-(x^2+y^2)^2 - 2(x^2-y^2) - 4xy - 1} \\ \\ s(x,y) = \dfrac{2((x^2+y^2)^2-1)}{-(x^2+y^2)^2 - 2(x^2-y^2) + 4xy - 1}. \end{cases}$$

Ces deux surfaces et les flots optiques associés $\phi_\omega + \frac{1}{r}\psi_v$, $\phi_{\omega'} + \frac{1}{s}\psi_{v'}$ sont représentés sur la figure (5.8).

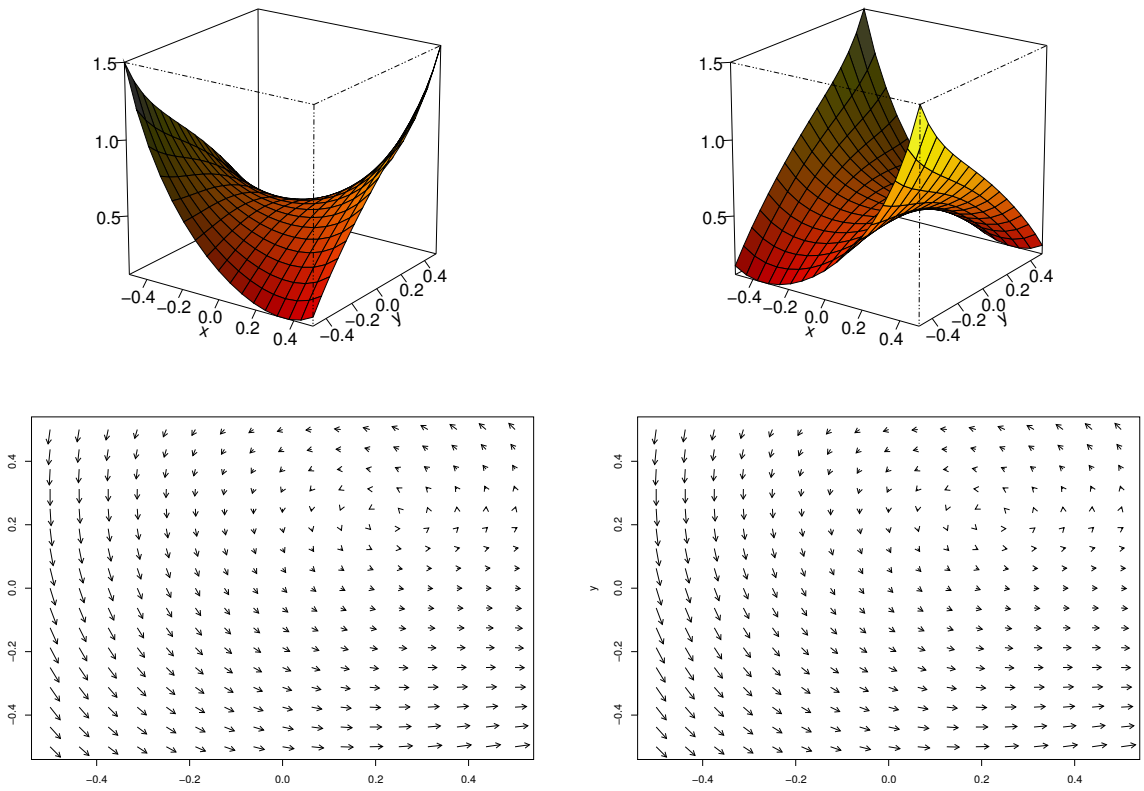

FIGURE 5.8: *En haut, à gauche, la surface* $1/r$ *et à droite la surface* $1/s$*. En bas, à gauche le flot généré par l'élément de* $\mathfrak{se}(3)$ $(\omega, v) = ((0,1,1),(-1,-1,0))$ *sur la surface* r *et à droite, le flot généré par l'élément de* $\mathfrak{se}(3)$ $(\omega', v') = ((0,0,1),(1,-1,0))$ *sur la surface* s*.*

Ceci nous mène immédiatement à la définition et au théorème suivants.

Définition 5.1 – *Soit U un sous-espace ouvert du disque D^2 et $\mathcal{A}(U)$ l'ensemble des fonctions de U dans $\mathbb{R}_+^*$ de la forme*

$$f(x,y) = \frac{N(x,y)}{D(x,y)}$$

avec

$$\begin{aligned} N(x,y) = \;& ((x^2+y^2)^2-1)\,(v_2v_1' - v_1v_2') + 2(x^2+y^2+1)\,[x(-v_3v_2' + v_2v_3') + y(v_3v_1' - v_1v_3')], \\ D(x,y) = \;& [(x^2+y^2)^2+1]\,(v_1'c_1 + v_2'c_2) + 2(x^2+y^2-1)\,[x(v_1'c_3 + v_3'c_1) + y(v_2'c_3 + v_3'c_2)] \\ & -4xy(v_1'c_2 + v_2'c_1) + 4(x^2+y^2)v_3'c_3 + 2(x^2-y^2)\,(-v_1'c_1 + v_2'c_2) \end{aligned}$$

où c, v et v' sont des vecteurs de $\mathbb{R}^3$ non nuls. On appelle surfaces ambiguës les fonctions de l'ensemble $\mathcal{A}(U)$.

Nous avons obtenu l'expression des surfaces ambiguës en inversant le système suivant

$$\begin{pmatrix} -\psi_v(m) & \psi_{v'}(m) \end{pmatrix} \begin{pmatrix} 1/r(m) \\ 1/s(m) \end{pmatrix} = \phi_c(m).$$

Théorème 5.4 – *Soient $\omega, \omega', v, v' \in \mathbb{R}^3$ et r, s deux fonctions positives définies sur $\mathbb{R}^2$. Si*

$$\forall m \in U, \;\; \phi_\omega(m) + \frac{1}{r(m)}\,\psi_v(m) = \phi_{\omega'}(m) + \frac{1}{s(m)}\,\psi_{v'}(m),$$

alors il existe $\tilde{r}, \tilde{s} \in \mathcal{A}(U)$ telles que $r|_U = \tilde{r}$ et $s|_U = \tilde{s}$.

Remarque – Les surfaces ambiguës sont ici décrites en termes de distance au centre optique de la caméra, en fonction des coordonnées des points projetés sur le plan $\{Z = 0\}$. Si on exprime ces surfaces en fonction des coordonnées des points dans $\mathbb{R}^3$, on obtient l'équation d'hyperboloïdes à une nappe, comme Horn dans [31] et Maybank dans [47].

5.5 Conclusion

Si on considère le flot optique dans un ensemble ouvert du plan $\{Z = 0\}$ contenant le disque ouvert D^2 (dans le cas de la projection sur la sphère suivie de la projection stéréographique), il n'y a aucune ambiguité sur le mouvement et les profondeurs de la scène qui l'ont généré. Mais si on observe le flot sur un ensemble ouvert strictement inclus dans D^2, il est possible qu'il soit ambigu. En pratique, par exemple dans le plan $\{Z = 1\}$ sur lequel la caméra réalise la projection dans le modèle sténopé, le flot optique pourra toujours être ambigu car on n'observera jamais le

plan tout entier. Si deux mouvements infinitésimaux sont donnés, on peut en déduire le domaine du plan $\{Z = 0\}$ ou $\{Z = 1\}$ (par les formules de projection d'un flot sur l'autre) sur lequel les flots observés ne seront jamais égaux et le domaine où l'ambiguité sera possible.

Cependant, l'ensemble des surfaces conduisant à l'ambiguité étant de mesure nulle, il est en pratique extrêmement rare de rencontrer un flot optique ambigu. Dans presque tous les cas, un flot optique exact porte suffisamment d'information pour que l'on puisse déterminer la rotation et, à une constante multiplicative près, la translation et les profondeurs de la scène filmée.

Bibliographie

[1] Azarbayejani, A., and Pentland, P. Recursive estimation of motion, structure and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 6 (1995), 562–575.

[2] Ballester, C., Caselles, V., and Verdera, J. Disocclusion by joint interpolation of vector fields and gray levels. *SIAM journal : "Multiscale Modelling and Simulation" 2*, 1 (2003), 80–123.

[3] Baron, J., Fleet, D., and Beauchemin, S. Performance of optical flow techniques. *International Journal of Computer Vision 12*, 1 (1994), 43–77.

[4] Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. Hierarchical model-based motion estimation. In *ECCV : Proceedings of the Second European Conference on Computer Vision* (1992), Springer-Verlag, pp. 237–252.

[5] Bertozzi, M., and Broggi, A. GOLD : A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing 7*, 62-81 (1998), 328–343.

[6] Black, M. *Robust incremental optical flow.* PhD thesis, Yale University, Departement of Computer Science, 1992.

[7] Brodsky, T., Fermuller, C., and Aloimonos, Y. Directions of motion fields are hardly ever ambiguous. *International Journal of Computer Vision 26* (1998), 5–24.

[8] Brooks, M., Chojnacki, W., and Baumela, L. Determining the ego-motion of an uncalibrated camera from instantaneous optical flow. *Journal of the Optical Society of America A 14*, 10 (1997), 2670–2677.

[9] Bruss, A., and Horn, B. Passive navigation. *Computer Graphics and Image Processing 21* (1983), 3–20.

[10] Cohignac, T., Lopez, C., and Morel, J.-M. Integral and local affine invariant parameter and application to shape recognition. In *Proceedings of International Conference on Pattern Recognition* (1994), pp. 164–168.

[11] Dibos, F. Du groupe projectif au groupe des recalages, une nouvelle modélisation. *Comptes Rendus de l'Académie des Sciences. Série I. Mathématique 332*, 9 (2001), 799–804.

[12] Dibos, F., Koepfler, G., and Monasse, P. *Image Alignment, in Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer-Verlag, 2003.

[13] Faugeras, O. *Three-dimensional computer vision : a geometric viewpoint*. MIT Press, 1993.

[14] Faugeras, O., Luong, Q., and Maybank, S. Camera self-calibration : theory and experiments. In *Proceedings of the Ninth IEEE International Conference on Computer Vision* (1992).

[15] Faugeras, O., Luong, Q., and Papadopoulo, T. *The Geometry of Multiple Images*. MIT Press, 2000.

[16] Faugeras, O., and Maybank, S. Motion from point matches : multiple of solutions. *International Journal of Computer Vision 4*, 3 (1990), 225–246.

[17] Felzenszwalb, P., and Huttenlocher, D. Efficient belief propagation for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition* (2004), vol. 1, pp. 261–268.

[18] Fusiello, A. Epipolar rectification. Tech. rep., Dipartimento Scientifico e Tecnologico, Università di Verona, 2000.

[19] Fusiello, A., Trucco, E., and Verri, A. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications 12*, 1 (2000), 16–22.

[20] Geman, S., and Geman, D. Stochastic relaxation, gibbs distribution, and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 6 (1984), 721–741.

[21] Gibson, J. The perception of the visual world. Tech. rep., Boston : Houghton Mifflin, 1950.

[22] Grammalidis, N., and Strintzis, M. Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences. *IEEE Transactions on Circuits and Systems for Video Technology 8*, 3 (1998), 328–343.

[23] Ha, J.-E., and Kweon, I.-S. Robust direct motion estimation considering discontinuity. *Pattern Recognition Letters 21*, 11 (2000), 999–1011.

[24] Hall, B. *Lie Groups, Lie Algebras, and Representations : An Elementary Introduction*. Springer-Verlag, 2003.

[25] Hartley, R. On defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 6 (1997), 580–593.

[26] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[27] Heeger, D., and Jepson, A. Subspace methods for recovering rigid motion i : algorithm and implementation. *International Journal of Computer Vision 7*, 2 (1992), 95–117.

[28] Helmholtz, H. *Treatise on Physiological Optics*. Dover, 1925.

[29] Hildreth, E. Recovering heading for visually-guided navigation. Tech. Rep. AIM-1297, MIT Artificial Intelligence Laboratory, 1991.

[30] Holland, P., and Welsh, R. Robust regression using iteratively reweighted least squares. *Communications in Statistics : Theory and Methods A*, 6 (1977), 813–828.

[31] Horn, B. Motion fields are hardly ever ambiguous. *International Journal of Computer Vision 1*, 10 (1987), 259–274.

[32] Horn, B., and Schunck, B. Determining optical flow. Tech. Rep. Memo 572, MIT, Artificial Intelligence Laboratory, 1980.

[33] Horn, B., and Weldon, E. Direct methods for recovering motion. *International Journal of Computer Vision 2* (1988), 51–76.

[34] Huang, T., and Faugeras, O. Some properties of the e matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 12 (1989), 1310–1312.

[35] Irani, M., and Anandan, P. About direct methods. In *Proceedings of the International Workshop on Vision Algorithms* (1999), pp. 267–277.

[36] Irani, M., Rousso, B., and Peleg, S. Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 3 (1997), 268–272.

[37] Jonchery, C., Dibos, F., and Koepfler, G. Décomposition de déformation pour l'estimation d'un mouvement de caméra. In *Actes du 20ème colloque GRETSI sur le traitement du signal et des images* (2005).

[38] Kanatani, K. *Geometric Computation for Machine Vision*. Oxford University Press, Inc., 1993.

[39] Kanatani, K. Renormalization for unbiased estimation. In *International Conference on Computer Vision* (1993), pp. 307–314.

[40] Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., and Tuceryan, M. Automated camera calibration and 3d egomotion estimation for augmented reality applications. In *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns* (1997), pp. 199–206.

[41] Levin, A., Zomet, A., and Weiss, Y. Learning to perceive transparency from the statistics of natural scenes. In *Advances in Neural Information Processing Systems* (2002), pp. 1247–1254.

[42] Longuet-Higgins, H. A computer algorithm for reconstructing a scene from two projections. *Nature 293*, 10 (1981), 133–135.

[43] Longuet-Higgins, H. The visual ambiguity of a moving plane. In *Proceedings of Royal Society of London* (1984), vol. 223, pp. 165–175.

[44] Luong, Q., Deriche, R., Faugeras, O., and Papadopoulo, T. On determining the fundamental matrix : analysis of different methods and experimental results. Tech. Rep. RR-1894, INRIA, 1993.

[45] Ma, Y., Kosecka, J., and Sastry, S. Linear differential algorithm for motion recovery : A geometric approach. *International Journal of Computer Vision 36*, 1 (2000), 71–89.

[46] Maybank, S. The angular velocity associated with the optical flow field due to a single moving rigid plane. In *Proceedings of Sixth European Conference of Artificial Intelligence* (1984), pp. 641–644.

[47] Maybank, S. *Theory of reconstruction from image motion*. Springer-Verlag, 1993.

[48] Monasse, P. Contrast invariant registration of images. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (1999), vol. 6, pp. 3221–3224.

[49] Needham, T. *Visual Complex Analysis*. Oxford University Press, 1997.

[50] Negahdaripour, S., and Horn, B. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*, 1 (1987), 168–176.

[51] Odobez, J., and Bouthémy, P. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation 6*, 4 (1995), 348–365.

[52] Ohm, J.-R. A realtime hardware system for stereoscopic videoconferencing with viewpoint adaptation. *Signal Processing : Image Communication 14* (1998), 147–171.

[53] Osborne, M. R. *Finite Algorithms in Optimization and Data Analysis*. John Wiley, New York, 1985.

[54] Osher, S., Rudin, L., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D 27*, 60 (1992), 259–268.

[55] Pearl, J. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[56] Ranchin, F. *Méthodes par ensembles de niveaux et modes conditionnels itérés pour la segmentation vidéo*. PhD thesis, Université Paris Dauphine, 2004.

[57] Rieger, J., and Lawton, D. Processing differential image motion. *Journal of Optical Society of America A 2* (1997), 354–360.

[58] Salamin, E. Application of quaternions to computation with rotations. Stanford Artificial Intelligence Lab, working paper, 1979.

[59] Sapiro, G., and Tannenbaum, A. Affine invariant scale-space. *International Journal of Computer Vision 11*, 1 (1993), 25–44.

[60] Scharstein, D., and Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision 47*, 1-3 (2002), 7–42.

[61] Schreer, O., Brandenburg, N., and Kauff, P. A comparative study on disparity analysis based on convergent and rectified views. In *Proceedings of the 11th British Machine Vision Conference* (2000), pp. 556–565.

[62] Sharon, D. Loopy belief propagation in image-based rendering. Tech. rep., Department of Computer Science, University of British Columbia, 2004.

[63] Shi, J., and Tomasi, C. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition* (1994), pp. 593–600.

[64] Soatto, S., Frezza, R., and Perona, P. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control 41*, 3 (1996), 393–414.

[65] Spindler, F. *Motion2D User Manual*.

[66] Subbarao, M., and Waxman, A. On the uniqueness of image flow solutions for planar surfaces in motion. In *Proceedings of IEEE Workshop on Computer Vision : Representation and control* (1985), pp. 129–140.

[67] Sun, J., Shum, H.-Y., and Zheng, N.-N. Stereo matching using belief propagation. In *Proceedings of the 7th European Conference on Computer Vision-Part II* (London, UK, 2002), Springer-Verlag, pp. 510–524.

[68] Tappen, M. F., and Freeman, W. T. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proceedings of the Ninth IEEE International Conference on Computer Vision* (2003), vol. 2, pp. 900–906.

[69] Tian, Y., Tomasi, C., and Heeger, D. Comparison of approaches to egomotion computation. In *IEEE Conference on Computer Vision and Pattern Recognition* (1996), pp. 315–320.

[70] Tomasi, C., and Shi, J. Direction of heading from image deformations. In *IEEE Conference on Computer Vision and Pattern Recognition* (1993), pp. 422–427.

[71] Tsai, R., and Huang, T. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 1 (1984), 13–27.

[72] Vassalo, R., Santos-Victor, J., and Schneebeli, H. A general approach for egomotion estimation with omnidirectional images. In *Proceedings of IEEE Workshop on Omnidirectional Vision* (2002), pp. 97–103.

[73] Vincent, E., and Laganière, R. Detecting planar homographies in an image pair. In *Proceedings of 2nd International Symposium on Image and Signal Processing and Analysis* (2001), pp. 182–187.

[74] Weickert, J., and Schnörr, C. Variational optic flow computation with a spatio-temporal smoothness constraint. Tech. rep., Computer Science Series, 2000.

[75] Weiss, Y. Bayesian belief propagation for image understanding. In *Workshop on Statistical and Computational Theories of Vision 1999* (1999).

[76] Weiss, Y., and Freeman, W. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory 47*, 2 (2001), 723–735.

[77] Yao, A., and Calway, A. Robust estimation of 3-d camera motion for uncalibrated augmented reality. Tech. rep., Dept of Computer Science, University of Bristol, 2002.

[78] Yedidia, J., Freeman, W., and Weiss, Y. Generalized belief propagation. In *Advances in Neural Information Processing Systems* (2000), pp. 689–695.

[79] Zakhor, A., and Lari, F. Edge based 3-d camera motion estimation with application to video coding. *IEEE Transactions on Image Processing 2*, 4 (1993), 481–498.

[80] Zucchelli, M., Santos-Victor, J., and Christensen, H. Constrained structure and motion estimation from optical flow. In *International Conference on Pattern Recognition* (2002), vol. 1, pp. 339–342.

Résumé : Cette thèse aborde le problème de l'estimation du mouvement d'une caméra filmant une scène fixe, à partir de la séquence d'images obtenue. La méthode proposée s'applique à l'estimation du mouvement entre deux images consécutives et repose sur la détermination d'une déformation 2D quadratique. À partir du mouvement estimé, nous étudions ensuite le problème de l'estimation de la structure de la scène filmée. Pour cela, nous appliquons une méthode de Belief Propagation directement sur un couple d'images, sans rectification, en utilisant l'estimation du mouvement. Enfin, nous examinons l'injectivité de la fonction associant un flot optique au mouvement d'une caméra et à la structure de la scène filmée. Deux mouvements de caméra étant donnés, nous décrivons le domaine d'observation où les flots générés sont susceptibles d'être identiques, et les surfaces filmées qui, associées aux deux mouvements, produiront ces flots ambigus.

Mots clés : mouvement, estimation, structure de la scène, Belief Propagation, flot optique, injectivité.

Abstract : This thesis deals with camera motion estimation, when the camera films a static scene, from the obtained sequence of images. The proposed method concerns motion estimation between two adjacent frames and is based on the determination of a 2D quadratic deformation between images. From the motion estimation, we next study the problem of scene structure estimation. We apply Belief Propagation method directly on an images couple, without any rectification, just using motion estimation. Finally, we study the injectivity of the map that associates an optical flow to camera motion and scene structure. Given two camera motions, we describe the domain where the two flows can be identical and the surfaces leading to these ambiguous flows.

Keywords : egomotion, estimation, scene structure, Belief Propagation, optical flow, injectivity.